

---

---

# Diseñando un gestor de energía de código abierto

---

---

Por  
Iván Fernández Mena y Alberto Pastor Moreno



**UNIVERSIDAD COMPLUTENSE  
MADRID**

Grado en Ingeniería del Software  
FACULTAD DE INFORMÁTICA

Jorge J. Gómez-Sanz  
Diseñando un gestor de energía de código  
abierto

MADRID, 2019–2020



---

---

# Designing an open source energy management system

---

---

By  
Iván Fernández Mena y Alberto Pastor Moreno



**UNIVERSIDAD COMPLUTENSE  
MADRID**

Software Engineering Degree  
COMPUTER SCIENCE FACULTY

Jorge J. Gómez-Sanz  
Designing an open source energy management  
system

MADRID, 2019–2020



# Autorización de difusión

Los abajo firmantes, matriculados en el Grado en Ingeniería del Software de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores el presente Trabajo Fin de Grado: 'Diseñando un gestor de energía de código abierto', realizado durante el curso académico 2019-2020 bajo la dirección de Jorge J. Gomez-Sanz en el Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Iván Fernández Mena y Alberto Pastor Moreno  
7 de Septiembre de 2020



# Dedicatorias

A mis padres y hermana por estar siempre ahí como soporte para mi ritmo de no parar. A mis amigos, por hacerme desconectar en los momentos difíciles y a Marina por apoyarme en todo momento. A todos, gracias.

- Iván Fernández Mena

A Guillermo y Daniel, por enseñarme a crecer y creer. A mis padres, por impulsar mi pasión. A mis amigos, por su apoyo incondicional. A Bego, por iluminar el camino. Muchas gracias, de corazón.

- Alberto Pastor Moreno





# Agradecimientos

A Jorge J. Gomez-Sanz por su apoyo, guía y paciencia durante todo el desarrollo del proyecto y a todas las personas que nos han apoyado durante todo este tiempo. A Daniel Pastor Moreno por su asesoramiento durante el desarrollo del modelo de optimización multiobjetivo.  
Muchas gracias.



# Resumen

La mejora de la eficiencia energética ha sido y es, un tema de fundamental importancia para la industria, el comercio y los gobiernos. Desde hace unos años se está poniendo empeño en lograr un nivel de eficiencia que resulte realmente significativo en todo el mundo y a todo nivel. Para lograrlo, es necesaria la disponibilidad de una herramienta libre y pública de gestión y monitorización de energía, comprendiendo el actual panorama tecnológico y energético. En este trabajo creamos una plataforma web que pretende facilitar este empeño, ofreciendo la información energética a aquellos perfiles de la Universidad Complutense de Madrid que tienen interés en analizar esta información para los diferentes centros en un espacio de tiempo determinado.

La mayoría de ocasiones en las que se necesita realizar estudios energéticos, se requiere observar una comparativa simulando la disposición de energías limpias; en el caso de este proyecto, nos centramos en la opción más viable actualmente, la energía fotovoltaica. El trabajo se separa en tres bloques principales: sección en la que se hace un tratamiento en tiempo real con tecnologías de *Big Data* de consumo estimado por ocupación de los centros, con sistemas de aviso si ésta comparativa descuadra. Otra sección provee de gráficas con parámetros configurables sobre el uso de energías renovables y cómo afecta al consumo de los centros. Por último, un modelo que predice el consumo energético según el consumo histórico de los centros, con opción de estipular un máximo de ese consumo y disponer de un aviso si ese consumo se podría superar.

## Palabras clave

EMS, energía fotovoltaica, análisis, optimización multiobjetivo, monitorización, Spark Structure Streaming



# Abstract

The improvement of energy efficiency has been and is, an issue of fundamental importance for industry, commerce and governments. For a few years now, efforts have been made to achieve a level of efficiency that is truly significant worldwide and at all levels. To achieve this, it is necessary to have a free and public tool for managing and monitoring energy, understanding the current technological and energy panorama. In this work we created a web platform that aims to facilitate this effort, offering energy information to those profiles of the Complutense University of Madrid who are interested in analyzing this information for different centers in a given time.

Most of the times when energy studies need to be carried out, it is required to observe a comparison simulating the availability of clean energies; in the case of this project, we focus on the most viable option at the moment: photovoltaic energy. The work is separated into three main blocks: section in which a treatment is made in real time with Big Data technologies of estimated consumption by occupation of the centers, with alarm systems if this comparative neglects. Another section provides graphs with configurable parameters on the use of renewable energy and how it affects the consumption of the centers. Finally, a model that predicts energy consumption according to the historical consumption of the centers, with the option of stipulating a maximum of that consumption and having a warning if that consumption could be exceeded.

## Keywords

EMS, photovoltaics, analysis, multiobjective optimization, monitoring, Spark Structure Streaming



# Índice general

Resumen	VII
Abstract	IX
	<b>Página</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Contexto . . . . .	1
1.2. Motivación . . . . .	2
1.3. Objetivos . . . . .	3
1.4. Estructura del documento . . . . .	4
<b>2. Introduction</b>	<b>5</b>
2.1. Context . . . . .	5
2.2. Motivation . . . . .	6
2.3. Objectives . . . . .	7
2.4. Document structure . . . . .	8
<b>3. Estado del arte</b>	<b>9</b>
3.1. Comunicando a través de la visualización de datos . . . . .	9
3.1.1. Energías renovables . . . . .	10
3.1.2. Edificios sostenibles . . . . .	11
3.1.3. Building Management System . . . . .	11
3.1.4. Opciones existentes . . . . .	11
3.1.5. Construcción de un Energy Management System . . . . .	13
3.2. Sistema de alarmas en tiempo real de consumo con estimaciones basadas en ocupación . . . . .	13
3.2.1. Posibles soluciones . . . . .	14
3.2.2. Nuestra propuesta . . . . .	17
3.3. Optimización presupuestaria multiobjetivo . . . . .	18
3.3.1. La Unión Europea demanda eficiencia energética . . . . .	18
3.3.2. Presupuestos energéticos multiobjetivo . . . . .	19
3.3.3. Algoritmia evolutiva . . . . .	21
3.3.4. Posibles soluciones . . . . .	23
3.3.5. Nuestra propuesta . . . . .	23
<b>4. Requisitos</b>	<b>25</b>
4.1. Plataforma web como base de un Energy Management System . . . . .	25
4.2. Representación de la información . . . . .	26
4.2.1. Datos de entrada de informes . . . . .	26

4.2.2.	Visualización gráfica . . . . .	27
4.3.	Optimización multiobjetivo . . . . .	28
4.3.1.	Formulación del problema . . . . .	29
4.3.2.	Ejemplos de optimización . . . . .	35
4.4.	Monitorización del consumo frente a la ocupación de un edificio . . . . .	39
4.4.1.	Respuesta eficiente en tiempo real . . . . .	40
<b>5.</b>	<b>Arquitectura software</b>	<b>41</b>
5.1.	Plataforma web . . . . .	42
5.1.1.	Componentes del sistema . . . . .	43
5.1.2.	Especificación del comportamiento . . . . .	45
5.2.	Fuentes de datos . . . . .	46
5.2.1.	Datos históricos de consumo de la Universidad Complutense de Madrid . . . . .	46
5.2.2.	Estimación de radiación fotovoltaica con web scraping . . . . .	47
5.2.3.	Simulación de datos en tiempo real . . . . .	50
5.3.	Monitorización en tiempo real con Spark Structured Streaming . . . . .	52
5.3.1.	Fases de la ETL . . . . .	53
5.4.	Optimización presupuestaria multiobjetivo . . . . .	56
5.4.1.	Detalles de implementación . . . . .	56
5.5.	Integración con Django . . . . .	57
5.5.1.	Diseñando para Django . . . . .	57
5.6.	Despliegue con contenedores Docker . . . . .	58
5.6.1.	Integración entre servicios . . . . .	59
5.6.2.	Gestión del despliegue . . . . .	59
<b>6.</b>	<b>Experimentación</b>	<b>61</b>
6.1.	Plataforma web solida . . . . .	61
6.2.	Análisis visual de consumo usando energía fotovoltaica . . . . .	62
6.2.1.	Formulario con configuración específica . . . . .	63
6.2.2.	Sección de consumo genérico . . . . .	64
6.2.3.	Sección de gráficas complejas . . . . .	67
6.3.	Configuración óptima usando energías renovables . . . . .	71
6.4.	Monitorización del consumo comparándolo con la ocupación de un edificio . . . . .	73
6.4.1.	Datos extraídos y simulados . . . . .	73
6.4.2.	Solicitud y vista general . . . . .	75
6.4.3.	Gráficas en tiempo real sin transformar . . . . .	77
6.4.4.	Indicadores de consumo y comparativas . . . . .	78
6.4.5.	Alarma de consumo con respecto a la ocupación esperada . . . . .	79
<b>7.</b>	<b>Conclusiones y trabajo futuro</b>	<b>81</b>
7.1.	Conclusiones . . . . .	81
7.2.	Trabajo futuro . . . . .	82
<b>8.</b>	<b>Conclusions and future work</b>	<b>85</b>
8.1.	Conclusions . . . . .	85
8.2.	Future work . . . . .	86
	<b>Índice de figuras</b>	<b>93</b>



---

<b>Índice de cuadros</b>	<b>95</b>
<b>Bibliografía</b>	<b>97</b>
<b>A. Código fuente</b>	<b>101</b>
A.1. Optimización multiobjetivo con JMetalPy . . . . .	101



# Capítulo 1

## Introducción

### 1.1. Contexto

Actualmente estamos atravesando una época de revolución de los datos. Esto está impulsado por, principalmente, las nuevas capacidades tecnológicas. Las tecnologías fundamentales que cambian la forma en la que reunimos, almacenamos, analizamos y transformamos la información han mejorado notoriamente en la última década. Paradigmas de desarrollo tecnológico como *Internet of Things* han conseguido encontrarle rentabilidad al tratamiento de los datos. Gracias a esto, múltiples industrias han puesto el foco sobre los datos, invirtiendo en su gestión y explotación, contribuyendo así a la mejora de las tecnologías involucradas.

La industria energética se ha visto beneficiada del uso de estas tecnologías. Al analizar sus datos, han conseguido ser más eficientes, ahorrando tiempo y dinero en la gestión energética. El uso de la información obtenida por el análisis es esencial en la toma de decisiones: permite reducir la incertidumbre y que, consecuentemente, se aumente la eficacia en el uso de la energía. Obtener beneficios de la gestión energética en la mayoría de organizaciones es una acción que está sujeta no solo al análisis de la energía que se consume, sino múltiples métricas externas. Algunas de estas métricas son la ocupación de los edificios, el uso porcentual de energías limpias y la disminución de la producción de  $CO_2$ . Para satisfacer todas estas métricas, es necesario considerar en los análisis muchos aspectos. Esta gran cantidad de factores que rodean a la gestión energética la convierten en un problema muy complejo pero muy relevante y necesario para la sociedad.

Un *Energy Management System*(EMS) es una herramienta software que permite supervisar, optimizar y gestionar el uso de energías de diferentes tipos (normalmente electricidad, gas o agua) de una instalación eléctrica. Una plataforma EMS facilita realizar seguimientos y análisis técnico de las instalaciones sobre las cuales ha sido instalado. El uso de un EMS está enfocado normalmente en identificar posibilidades de ahorro y eficiencia para mejorar el uso energético. Los EMS le otorgan beneficios claves a una organización, debido a que éstas, normalmente, emplean mucho esfuerzo en obtener conocimiento que resulte relevante en su gestión energética. Beneficios como la capacidad de tomar decisiones respaldadas por los datos o de optimizar el consumo energético para reducir desperdicios, son algunos de los más importantes. Una gestión energética eficiente conduce a mejoras tanto económicas como medioambientales.

Es también importante destacar la necesidad de un EMS para la superación de un certificado de gestión energética como es el ISO 50001. Este estándar implanta una política energética que facilita la identificación y priorización de los aspectos energéticos de una organización en pos de la eficiencia. Evaluar el cumplimiento de los requisitos legales o establecer objetivos de mejora en la optimización son algunos de los aspectos del ISO 50001 que se pueden tratar con un EMS[1]. Además, los análisis realizados en un EMS ayudan a las auditorías externas y a establecer procedimientos de control y seguimiento. ISO 50001 busca la mejora continua del uso energético, algo que un EMS facilitará en gran medida.

## 1.2. Motivación

Actualmente, las organizaciones disponen de una cantidad muy elevada de fuentes de datos relacionados a su gestión de la energía eléctrica. Estas fuentes extraen datos de las instalaciones energéticas y los profesionales responsables de éstas son los principales usuarios de nuestra plataforma software. Su principal interés en el uso de un EMS es obtener valor de sus datos y esto lo pueden conseguir realizando análisis para reducir la incertidumbre en la toma de decisiones sobre su uso energético. Sin la inclusión de un sistema como el que planteamos, éstos profesionales requieren de estudios complejos, específicos y laboriosos. Esto sería contraproducente debido a que los técnicos responsables de las instalaciones comparten métricas e intereses de análisis comunes. Esta situación se resuelve con el desarrollo de un EMS de código abierto suficientemente potente.

Al igual que el uso de información específica de energías limpias, existen otros mecanismos que son tratados para tener una gestión completa de un sistema energético. Un sistema de alarmas y monitorización es una parte fundamental de un EMS; conocer si el uso energético es el esperado en tiempo real y establecer métricas específicas, ayuda a controlar posibles problemas de consumo. Para ello, la ocupación de los edificios ofrece una estimación precisa sobre si el consumo es real y proporcional al esperado. Si existiera mucha diferencia, se alertaría del incidente al instante de que ocurriera. La facturación energética tiene un periodo muy largo, por lo que un error en la configuración de la instalación podría tardar mucho en ser identificada por sus responsables. Utilizando un sistema de supervisión en tiempo real acortamos enormemente este periodo, ya que la identificación de anomalías se produciría en el mismo instante en el que ocurren. Esto daría lugar a grandes ahorros, ya que las configuraciones energéticas y el uso de las instalaciones se podrían adaptar y ajustar a la necesidad verdadera de los usuarios de éstas en tiempo real.

Un uso energético incorrecto por parte de una institución podría suponer un impacto medioambiental negativo. El certificado de la normativa internacional ISO 50001, que tiene como objetivo mantener y mejorar un sistema de gestión de energía en una organización, establece lo necesario para que las organizaciones mejoren continuamente la eficiencia, los costes relacionados con el uso de la energía y la emisión de gases de efecto invernadero. Cumplir la ISO 50001 muestra una responsabilidad real ante la mejora continua del uso energético que tiene una organización, de forma nacional e internacional, ya que la ISO 50001 compone cuatro (7, 11, 12 y 13) de los *Sustainable Development*

*Goals* establecidos por la Unión Europea dentro de la Agenda de desarrollo sostenible. La utilización de un EMS proporciona una base sólida para obtener una certificación energética de este tipo y así cumplir con los objetivos de resiliencia energética.

### 1.3. Objetivos

Este proyecto tiene como principal objetivo desarrollar un *Energy Management System* que se constituya como una solución de software libre al problema de la gestión energética. Nuestra propuesta facilita la gestión y monitorización de la energía eléctrica a través del tratamiento y explotación de datos. Este uso de los datos permitirá realizar análisis energéticos exhaustivos de instalaciones. Junto al uso de datos de estimaciones de uso de energía solar, datos de ocupación de los centros y técnicas de optimización multiobjetivo, nuestro EMS facilitará, entre otros aspectos, la toma de decisiones pertinentes a instalaciones energéticas. De esta forma, la plataforma ayuda a que más organizaciones tengan acceso a una herramienta de este tipo. Además, al ser software libre, permite que ocurra el desarrollo colaborativo. Gracias a esto, el sistema mejoraría con las contribuciones de los usuarios.

Otro de los objetivos del proyecto es cumplir en lo máximo posible con la necesidad de la Universidad Complutense de Madrid de disponer de una forma sencilla y unificada de analíticas energéticas de los centros de los que dispone. Dado que se espera que el sistema sea accedido por todo tipo de perfiles, expertos en el uso de las nuevas tecnologías o no, se ha dispuesto de una plataforma web sencilla en interacción y usabilidad, sin necesidad de manuales ni guías de usuario.

En el desarrollo de la plataforma EMS hemos buscado poder proporcionar las siguientes posibilidades a los usuarios:

- **Conocer el comportamiento del consumo energético:** Los datos de los que disponen los responsables de las instalaciones necesitan de un tratamiento que, si fuera manual, tomaría demasiado tiempo y costes. Por ello, realizar pre-procesamiento de estos datos en crudo y mostrarlos en gráficas ricas agregando datos de forma automática, le aporta mucho valor a un EMS. De esta forma, se consigue conocer el comportamiento del consumo. Además, los cruces de información de consumo de instalaciones con una instalación hipotética de energías limpias, en particular de energía fotovoltaica, sería inviable si no fuera realizado con un sistema como el que planteamos. Por esto, este proyecto ofrece, además, el poder indagar en el impacto que supondría el uso de esta fuente de energía verde en el consumo de una instalación.

El apoyo de la energía fotovoltaica impacta directamente en la reducción de costes, convirtiendo a un centro en más autónomo. Además, su consumo será más responsable con el medio ambiente. Es por esto por lo que la visualización de datos temporales agregados incluyendo la energía fotovoltaica tiene un gran peso en nuestra plataforma.

- **Calcular configuraciones óptimas de instalaciones fotovoltaicas:** La reducción de costes asociados al consumo no se puede realizar de forma trivial y directa; los obstáculos que encontramos a la hora de diseñar la configuración energética de

una instalación requieren la optimización simultánea de más de un objetivo. En el proyecto se tienen en cuenta factores como la reducción al máximo del consumo de electricidad de la red con un coste de inversión mínimo y el uso de la mejor configuración posible de energía solar. De esta forma, incluimos energía solar siempre y cuando reduzca los costes, considerando factores como la vida media de las baterías, el mantenimiento de éstas o el coste de instalación. Introducir varias métricas diferentes puede generar conflictos entre objetivos, que harán que la mejora de uno de ellos dé lugar a un empeoramiento de algún otro. Por eso, en el proyecto un objetivo importante es el uso de técnicas de optimización multiobjetivo que hagan llegar a una situación de compromiso en la que todos los objetivos sean satisfechos en un grado aceptable, dando lugar a conocimiento útil en la toma de decisiones presupuestarias.

- **Monitorizar el consumo en tiempo real:** Utilizando un sistema de avisos, es relevante analizar si el consumo en cada momento supera al consumo esperado. Si fuera así, sería posible conocer en tiempo real si la configuración de una instalación ha alterado el consumo negativamente. Dado que el consumo está asociado al uso de las instalaciones por parte de los usuarios, es relevante conocer la ocupación del centro. Así, el sistema puede estipular con más precisión qué es considerado como normales. Además, también permite realizar estudios de tendencias con datos del consumo asociado a la ocupación de un edificio en tiempo real según se obtengan de origen.

## 1.4. Estructura del documento

La memoria estará dividida en los siguiente apartados:

1. **Introducción:** Breve resumen del proyecto, motivaciones y objetivos de este.
2. **Estado del arte:** Descripción del estado actual de las tecnologías utilizadas para este proyecto.
3. **Requisitos:** Explicación de los servicios que el proyecto debe ofrece para cumplir con lo esperado.
4. **Arquitectura y componentes:** Contiene un resumen de la arquitectura del proyecto, los componentes utilizados y la interacción entre ellos.
5. **Experimentación:** Muestra el resultado del proyecto, así como las pruebas realizadas.
6. **Conclusiones:** Conocimiento y conclusiones que hemos podido extraer de nuestro trabajo, además de posibles maneras de continuarlo o mejorarlo en un futuro.

# Capítulo 2

## Introduction

### 2.1. Context

We are currently going through a time of data revolution. This is mainly driven by new technological capabilities. Fundamental technologies that change the way we collect, store, analyze and transform information have improved dramatically in the last decade. Technological development paradigms such as *Internet of Things* have managed to find profitability in data processing. Thanks to this, many industries have put the focus on data, investing in its management and exploitation, thus contributing to the improvement of the technologies involved.

The energy industry has benefited from the use of these technologies. By analyzing their data, they have become more efficient, saving time and money in energy management. The use of the information obtained by the analysis is essential in the decision making process: it allows reducing the uncertainty and, consequently, increasing the efficiency in the use of energy. Obtaining benefits from energy management in most organizations is an action that is subject not only to the analysis of the energy that is consumed, but multiple external metrics. Some of these metrics are the occupation of buildings, the percentage use of clean energy and the decrease in production of  $CO_2$ . To satisfy all these metrics, many aspects need to be considered in the analysis. This great amount of factors that surround the energy management makes it a very complex problem but very relevant and necessary for the society.

An *Energy Management System* (EMS) is a software tool that allows you to monitor, optimize and manage the use of different types of energy (usually electricity, gas or water) in an electrical installation. An EMS platform facilitates the monitoring and technical analysis of the installations on which it has been installed. The use of an EMS is normally focused on identifying possibilities of savings and efficiency to improve energy use. EMSs provide key benefits to an organization, since they usually spend a lot of effort in obtaining knowledge that is relevant to its energy management. Benefits such as the ability to make data-supported decisions or to optimize energy consumption to reduce waste are some of the most important. Efficient energy management leads to both economic and environmental improvements.

It is also important to highlight the use of an EMS for the achievement of an energy

management certificate such as ISO 50001. This standard establishes an energy policy that facilitates the identification and prioritization of the energy aspects of an organization in pursuit of efficiency. Assessing compliance with legal requirements or establishing improvement objectives in the optimization are some of the aspects of ISO 50001 that can be dealt with an EMS. In addition, the analyses performed in an EMS help external audits and establish control and monitoring procedures. ISO 50001 seeks continuous improvement in energy use, something that an EMS will greatly facilitate.

## 2.2. Motivation

Currently, organizations have a very high amount of data sources related to their electrical energy management. These sources extract data from energy facilities. The professionals responsible for these are the main users of our software platform. Their main interest in the use of an EMS is to obtain value from their data. They can achieve this by performing analyses to reduce the uncertainty in decision making about their energy use. Without the inclusion of a system like the one we propose, these professionals require complex, specific and laborious studies. This would be counterproductive because the technicians responsible for the facilities share common metrics and analysis interests. This situation is solved with the development of a sufficiently powerful open source EMS.

As well as the use of specific clean energy information, there are other mechanisms that are dealt with to have a complete management of an energy system. An alarm and monitoring system is a fundamental part of an EMS; knowing if the energy use is the expected one in real time and establishing specific metrics, will help to control possible consumption problems. To this end, the occupation of the buildings provides an accurate estimate of whether the consumption is real and proportional to the expected one. If there were a big difference, the incident would be alerted the moment it occurred. Energy billing has a very long period, so an error in the configuration of the installation could take a long time to be identified by those responsible. By using a real-time monitoring system, this period is greatly shortened, since the identification of anomalies would occur at the very moment they occur. This would result in great savings, since the energy configurations and the use of the installations could be adapted and adjusted to the real need of the users of these installations in real time.

Incorrect energy use by an institution could have a potentially negative environmental impact. The ISO 50001 international standard certificate, which aims to maintain and improve an energy management system in an organization, establishes what is necessary for organizations to continuously improve efficiency, costs related to energy use and greenhouse gas emissions. Complying with ISO 50001 shows a real responsibility for the continuous improvement of an organization's energy use, both nationally and internationally, since ISO 50001 is part of four (7, 11, 12 and 13) of the *Sustainable Development Goals* established by the European Union within the Sustainable Development Agenda. The use of an EMS provides a solid basis for obtaining an energy certification of this type and thus meet the objectives of energy resilience.



## 2.3. Objectives

The main objective of this project is to develop an *Energy Management System* as a free software solution to the problem of energy management. Our proposal facilitates the management and monitoring of electrical energy through the processing and exploitation of data. This use of the data will allow to make exhaustive energetic analyses of facilities. Together with the use of data from solar energy use estimations, center occupation data and multiobjective optimization techniques, our EMS will facilitate, among other aspects, decision making relevant to energy installations. As it is free software, the organizations that use it will not have to incur costs. In this way, the platform helps more organizations to have access to a tool of this type. In addition, it allows collaborative development to occur. Thanks to this, the system would be improved with the contributions of the users.

Another objective of the project is to meet as much as possible the need of the Complutense University of Madrid to have a simple and unified form of energy analysis of the centers it has. Since it is expected that the system will be accessed by all types of profiles, experts in the use of new technologies and others, a simple web platform in terms of interaction and usability has been made available, without the need for manuals or user guides.

In the development of the EMS platform we have sought to provide the following possibilities to users:

- **Knowing energy consumption behavior:** The data available to those responsible for the facilities need to be processed, which, if it had to be manual, would take too much time and cost. Therefore, pre-processing this raw data and displaying it in rich graphs by automatically adding data, adds a lot of value to an EMS. In this way, it is possible to know the consumption behavior. In addition, the cross-checking of consumption information from installations with a hypothetical clean energy installation, particularly photovoltaic energy, would be unfeasible if it were not done with a system like the one we are proposing. Therefore, this project also offers the possibility of studying the impact that the use of this green energy source would have on the consumption of an installation.

The support of photovoltaic energy has a direct impact on the reduction of costs, making a center more autonomous. In addition, its consumption will be more responsible with the environment. This is why the visualization of aggregated temporal data including photovoltaic energy has a great weight in our platform.

- **Calculate optimal configurations of photovoltaic installations:** Reducing costs associated with consumption cannot be done in a trivial and direct way; the obstacles we encounter when designing the energy configuration of an installation require the simultaneous optimization of more than one objective. The project takes into account factors such as the maximum reduction of electricity consumption from the grid with a minimum investment cost and the use of the best possible configuration of solar energy. In this way, we include solar energy as long as it reduces costs, considering factors such as the average life of the batteries, their maintenance or the cost of installation. Introducing several different metrics can generate conflicts between objectives, which will make the improvement of one of them lead to a worsening of some other. Therefore, an important objective in the project is the use of multiobjective optimization techniques that will lead to a

compromised situation in which all objectives are satisfied to an acceptable degree, resulting in useful knowledge in financial decision making.

- **Monitoring consumption in real time:** It is relevant to analyze whether the consumption at each time exceeds the expected consumption by using an alarm system. If so, it would be possible to know in real time if the configuration of an installation has altered consumption negatively. Given that consumption is associated with the use of facilities by users, it is relevant to know the occupation of the center. Thus, the system can stipulate more precisely what is considered normal. In addition, it also allows for studies of time trends in consumption associated with the occupation of a building.

## 2.4. Document structure

The report will be divided into the following sections:

1. **Introduction:** Brief summary of the project, motivations and objectives.
2. **State of the art:** Description of the current state of the technologies used for this project.
3. **Requirements:** Explanation of the services that the project must offer to meet the expectations.
4. **Architecture and components:** Contains a summary of the project's architecture, the components used and the interaction between them.
5. **Experimentation:** Shows the result of the project, as well as the tests performed.
6. **Conclusions:** Knowledge and conclusions that we have been able to extract from our work, as well as possible ways to continue or improve it in the future.

# Capítulo 3

## Estado del arte

Durante el desarrollo del proyecto, destacamos tres bloques tecnológicos que sustentan los objetivos principales del proyecto: la plataforma web donde se le muestra al usuario todas las funcionalidades de la aplicación y el contenido generado por ésta, el algoritmo evolutivo que, dada una localización y demanda prevista, diseña una configuración fotovoltaica óptima y el sistema de monitorización en tiempo real que compara el consumo estimado con la ocupación de los edificios. A continuación expondremos la base teórica sobre la cual se sustenta cada uno de estos bloques.

### 3.1. Comunicando a través de la visualización de datos

Gracias al auge de los datos y al conocimiento de las empresas y las instituciones de que se puede obtener mucho valor de los datos, la explotación de éstos a través de análisis computacional ha llegado a la industria energética. Durante décadas, los análisis requeridos para estudiar el comportamiento del mercado, por parte de las compañías energéticas, o del consumo propio, por parte de compradores de energía, conllevaba estudios muy lentos, pesados y costosos. En cambio, actualmente, la tecnología posee la capacidad de desarrollar herramientas digitales que les permiten ver a ambas partes al instante gráficas muy elaboradas, las cuales requerirían demasiados cálculos para ser procesadas manualmente.

La visualización de datos se ha convertido en una actividad esencial para cualquier organización energética, ya que nos permite contrastar, comparar e interpretar los datos, obteniendo así un conocimiento profundo que no pudiera ser obtenido a simple vista. El cruce entre múltiples fuentes de datos o entre ventanas temporales diferentes, entre otras prácticas comunes en la visualización de datos, nos permiten mirar a los datos desde un enfoque que nos puede ayudar a entender mejor el comportamiento de estos. La visualización supone un paso más en el análisis de los datos, aportándole recursos al experto que sabe cómo interpretar las diferentes gráficas o dibujos que se le presentan.

En el dominio de la industria energética, los datos provienen de sensores y contadores, entre otros elementos, alojados en las instalaciones eléctricas. Desde estos componentes se recogen datos en crudo, los cuales necesitan un preprocesamiento previo a la visualización. En este preprocesamiento se limpian los datos, excluyendo aquellos que sean

inservibles, se ordenan, se normalizan y se formatean coherentemente para que todos los datos puedan ser tratados por igual en la siguiente fase. Tras esto, se construyen gráficas y/o tablas, dependiendo del caso de uso y de la necesidad analítica.

El resultado de la visualización de datos se le presenta al responsable de la instalación, el cual puede aplicar sus conocimientos sobre el dominio energético para interpretarlos coherentemente. La visualización de datos transforma los datos para que los responsables puedan comprenderlos y explotarlos fácilmente. Gracias a esto, ellos podrán basar su toma de decisiones en los datos, reduciendo la incertidumbre del proceso.

Nuestro proyecto tiene como objetivo la construcción de una plataforma que incluya, entre otras funcionalidades que explicaremos posteriormente, visualización de datos del contexto energético. Este proceso incluye las transformaciones pertinentes para recoger datos en crudo y devolver gráficas que agreguen múltiples fuentes de datos y utilicen ventanas temporales. Estos datos se recogen en las instalaciones energéticas y se desean explotar por sus responsables a través de ordenadores de escritorio.

### 3.1.1. Energías renovables

En muchas de las ocasiones, aunque se disponga de información energética suficiente, como datos sobre el histórico de consumo, la potencia utilizada, la energía reactiva y el  $CO_2$  producido, es necesario el uso de agentes externos para reducir el consumo. El agente principal es el uso de energías limpias. A día de hoy, el uso de energías verdes está cogiendo fuerza, vivimos en un momento en el que las organizaciones necesitan es ser lo mas autosuficientes posibles y reducir su impacto al medio ambiente. El hecho de añadir una fuente de energía nueva requiere de un esfuerzo añadido de análisis previo y un posterior seguimiento, diferente en cada centro. Proporcionar a los técnicos la información necesaria sobre diferentes configuraciones es esencial para que ellos puedan solicitar la instalación más oportuna para cada centro. Esta situación le brinda mucha importancia al uso de un EMS para tratar de generar este conocimiento tan necesario.

Una de las mayores problemáticas del panorama energético, dada la entrada de energías renovables en el mercado, es conocer qué rentabilidad y eficiencia pueden aportarles a un sistema energético. En la resolución de este problema se encuentran varios factores y obstáculos que deben ser tenidos en cuenta. Si estos fueran omitidos, los resultados no serían suficientemente precisos. Integrar energías limpias en un sistema no es sencillo, ya que modifica enormemente cómo se presupuesta la energía en una instalación. De esta forma, incluir las energías limpias en los análisis de un EMS facilita todas las gestiones pertinentes a éstas. Debido al apogeo de la industria energética verde y la necesidad medioambiental de la inclusión de este tipo de energías, esta inclusión es cada vez más urgente y necesaria.

En la inclusión de energías renovables en nuestro proyecto consideramos únicamente el uso de energía solar fotovoltaica, la cual se basa en el aprovechamiento de la radiación solar, ya que ésta permite instalar sistemas pequeños fácilmente escalables. Además, una instalación fotovoltaica puede ser conectada a una red eléctrica ya existente, algo indispensable en nuestro caso de uso. Por último, la energía solar fotovoltaica es la opción

más eficiente económicamente de auto-consumo energético en el presente.

### 3.1.2. Edificios sostenibles

Los sistemas de gestión de la energía se llevan usando desde hace tiempo en aplicaciones en las que la electricidad es vital, como industria, *data centers* y hospitales, pero desde la pasada década, podemos considerar como comenzada una nueva época donde la importancia de la gestión energética eficiente y verde está presente en cualquier ámbito. En todo tipo de edificios, desde edificios pertenecientes a la administración pública, hasta edificios residenciales u oficinas privadas, podemos ver cómo la gestión de los recursos energéticos ha cobrado una gran importancia. Los gestores de mantenimiento y asesores energéticos se encuentran ante el problema de convertir un espacio que se basa en su consumo de energía tradicional, hacia un edificio que utiliza energías limpias y renovables de manera inteligente y controlada.

### 3.1.3. Building Management System

Un *Building Management System* es un software que permite realizar todas las gestiones necesarias de un edificio con el fin de que se vuelva un 'edificio inteligente'. Dentro de estas gestiones, las más comunes suelen ser:

1. **Control y centralización** de la mayor cantidad posible de elementos y aspectos del edificio.
2. **Mantenimiento preventivo** de las estructuras del edificio.
3. **Información y análisis** de los consumos energéticos del edificio.
4. **Seguridad** de los edificios, protegiendo tanto a las personas que lo transiten como los bienes materiales que se encuentren en éste.
5. Uso de **técnicas predictivas** de *Machine Learning* o *Deep Learning*, en el contexto *Big Data*, para la extracción de valor desde datos relacionados con el edificio, prediciendo consumos y costes energéticos.

En el caso de nuestro estudio, nos centraremos en el punto 3 y punto 5, envueltos por el contexto de la monitorización energética.

Nuestra plataforma permite la obtención de diferentes informes que muestran resultados analíticos con el fin de proporcionar ayuda a los diferentes *stakeholders* del sistema. De esta forma, pueden conocer aspectos no triviales sobre el estado energético del edificio. Así, nuestra plataforma proporciona una base de información sobre la cual los técnicos del sistema puedan tomar decisiones adecuadas.

### 3.1.4. Opciones existentes

La plataforma que desarrollamos en este proyecto está compuesta por algunas de las funcionalidades que normalmente encontraríamos en un *Energy Management System*, como por ejemplo, la visualización de datos[2]. Aun así, los sistemas EMS inciden más en

el control, entre otros aspectos.

Un *Energy Management System* debe proveer de las siguientes funciones según el ISO 50001[1][3]:

1. **Monitorización en tiempo real:** Permite detectar incidencias en cualquier momento, sin esperar a que se consuma un intervalo de tiempo.
2. **Análisis de datos históricos:** Muestra información a través de analizar datos pasados con el objetivo de descubrir tendencias y patrones.
3. **Análisis de la calidad de la instalación:** Evalúa la calidad y el estado de los componentes energéticos.
4. **Integración con sensores:** Consume datos de sensores ubicados en instalaciones eléctricas.
5. **Informes personalizados:** Presenta información con las necesidades de visualización de datos pertinentes al conjunto de los usuarios de la plataforma.
6. **Facturación automatizada:** Permite operar sobre la facturación del sistema sin necesidad de la interacción de un usuario.

En el ámbito de software EMS privativo encontramos *Wattics*, *EnergyCAP* y *EcoStructure*. Estas son algunas de las herramientas de software más utilizadas actualmente para la gestión energética a gran escala. Estas herramientas cuentan con la capacidad para escalar decentemente ante grandes cantidades de datos homogéneos a través del paradigma de IoT(*Internet of Things*). Con este enfoque presente, disponen de la capacidad de realizar análisis de datos con cierta facilidad. En cambio, son productos que tienen un gran costo, identifican casos de uso ingenuamente genéricos y carecen de flexibilidad para el despliegue.

La aplicación EMS *open-source* más popular, que podría haber sido candidata a ser usada para este caso de uso, es *OpenEMS*. *OpenEMS* es una aplicación modular de gestión de sistemas energéticos que permite controlar y monitorizar componentes como generadores o baterías. Incluye una arquitectura mucho más flexible y potente que las herramientas previamente discutidas.

*OpenEMS* carece de las funcionalidades más importantes que desarrollamos en este proyecto, como la monitorización en tiempo real. Además, su despliegue requiere de una gran cantidad de recursos, ya que el sistema debe ser ejecutado al completo: todos los dispositivos partícipes, tanto sensores como proyectores de los resultados analíticos, deben estar encendidos para que la plataforma funcione correctamente. Dada su arquitectura, el procesamiento de datos posee una gran latencia. Además, imposibilita agregar datos de diferentes fuentes. Por último, carece de casos de uso enfocados en el ámbito de la energía fotovoltaica y de analítica compleja.

### 3.1.5. Construcción de un Energy Management System

La plataforma que desarrollamos en este proyecto se apoya sobre la visualización de datos, pero lo integra con otras dos funcionalidades clave en el contexto energético, como son el diseño de configuraciones óptimas de instalaciones eléctricas y la monitorización del consumo real contra el consumo estimado. A pesar de que las alternativas que hemos comentado en el apartado anterior están compuestas de varias funcionalidades, éstas no se encuentran entre ellas. Esto deja un hueco en el panorama de software EMS para el desarrollo de una herramienta que incluya estas funcionalidades. Poder interactuar con los datos de múltiples formas en la misma plataforma facilita la integridad y, consecuentemente, la operabilidad.

Nuestro proyecto se enfoca en la gestión energética de edificios, tomando aspectos de un *Building Management System*, dentro del contexto global de un *Energy Management System* con licencia de software libre. Esta composición de funcionalidades no se da en ninguna herramienta de las consideradas como alternativas. Nuestra plataforma facilitará el trabajo de todos los *stakeholders* de las instalaciones eléctricas, permitiéndoles ver de un vistazo la actividad energética de cada instalación. Para aumentar la usabilidad y explotar la facilidad de uso, desarrollamos la plataforma como una aplicación web, ya que los usuarios para los cuales está destinada trabajan a diario desde un ordenador y, por lo tanto, les será más cómodo integrar su uso en su flujo de trabajo diario.

## 3.2. Sistema de alarmas en tiempo real de consumo con estimaciones basadas en ocupación

Un sistema de alarmas y monitorización es un pilar fundamental dentro de un *Energy Management System* debido a la posibilidad de ofrecer a un perfil técnico una forma sencilla de controlar el funcionamiento de un sistema energético a partir de unas métricas especificadas previamente. La utilización de este tipo de herramientas le ofrece ser capaz de prever incidencias y detectar problemas de forma rápida, ahorrando tiempo y costes, además de mejorar la eficacia y eficiencia de las labores de mantenimiento.

Usando la monitorización de consumos en un sistema energético, podemos obtener los siguientes beneficios:

- **Medición de parámetros clave** de una instalación como la potencia demandada y la energía consumida (activa y reactiva).
- **Detección de incidencias**, como sobretensiones, posibles interrupciones, trabajos en vacío en los centros y cuándo los centros están trabajando con rendimientos inferiores a lo esperado.
- **Estudio del consumo** de cada tipo de centro monitorizado, lo cual puede conllevar una mejora de la productividad a través de la obtención de indicadores energéticos y económicos.
- **Establecer la línea base de consumos** y calcular los ahorros en un contrato de servicios energéticos.

- Permite la **adquisición de lecturas de múltiples indicadores medioambientales**.
- Se pueden programar **alarmas que notifiquen las incidencias** en el consumo de forma instantánea.
- **Emisión de informes de eficiencia energética** con diagnósticos y propuestas de actuaciones para el ahorro.
- **Formar a los empleados en las buenas prácticas energéticas**, de forma que adecuen sus hábitos de consumos en base a datos objetivos obtenidos con la monitorización.

### 3.2.1. Posibles soluciones

En la implementación de nuestra solución, existen varias posibilidades que deben ser consideradas.

#### Scripting

La primera opción es la más básica a nivel arquitectónico. Se trata de la creación de una pequeña aplicación con un lenguaje de programación de *scripting*. Esta aplicación debe ser capaz de la utilización de protocolos y librerías de comunicación por red para enviar información cada vez que reciba datos. Este programa, al ser implementado desde cero, se puede ajustar mucho más al problema. Además, el uso de un programa específico permite que la integración con nuestra plataforma se pueda realizar de forma eficaz. Esto se debe a la propia flexibilidad de la creación desde cero del programa y con éste los conectores necesarios.

En la aplicación desarrollada habría que tener en cuenta diferentes factores que obstaculizan la resolución del problema. El control de fallos o caídas del sistema, de forma que no se pierda información en el procesamiento, es uno de estos factores. También es importante el proveer de seguridad al sistema de datos, generando una capa de seguridad o cubriendo los datos ante una posible intrusión no autorizada. Esto es esencial en una plataforma como la nuestra que dispone de datos de índole no pública, es decir, que requieren de acceso autorizado. Por último, sería necesario gestionar la carga simultánea de datos que se va a procesar en el sistema, además de la naturaleza del procesamiento, si es distribuido y paralelo, detalle que puede ser esencial [4].

Considerando los detalles a tener en cuenta, determinamos que no es una solución viable para este proyecto. El proyecto dispone de un gran número de detalles que obstaculizan el desarrollo de la funcionalidad a través de esta solución. Es posible su uso, pero a nivel tecnológico existen opciones que se ajustan más a los objetivos establecidos.

#### Herramientas de procesamiento

Las herramientas de procesamiento *Big Data* son idóneas para cubrir los problemas del *scripting*. Además, muchas de estas tecnologías ofrecen una amplia gama de funcionalidades útiles para realizar todas las transformaciones exigidas por el problema, así como



realizar conexiones entre elementos. Estas funcionalidades son diseñadas para poder escalar eficientemente en horizontal, por lo que aportan mucha flexibilidad[4].

Al contrastar los objetivos del proyecto con lo que nos ofrece una tecnología de procesamiento *Big Data*, comprobamos que es la opción más razonable a usar. Estas herramientas nos proveen de una base sólida con gran capacidad de amplitud para el desarrollo del caso de uso específico. Como contraposición, debemos tener en cuenta que el uso de una de estas tecnologías no es trivial. Es importante gestionar cómo tratan estas tecnologías las tareas, el estado global de los datos y cómo usar sus capacidades para sacar el mayor potencial a los datos.

Existen tecnologías en este dominio muy consolidadas en el mercado tecnológico, especializadas en el tratamiento y procesamiento de datos en tiempo real [5]. Las principales son las siguientes:

- **Apache Samza:** Es un *framework* de procesamiento distribuido de código abierto creado por *LinkedIn* para resolver varios tipos de requisitos de procesamiento en tiempo real, como seguimiento de la información, registro y gestión de los datos y sistemas de ingesta de datos en tiempo real. *Samza* está diseñado para manejar mensajes grandes y poder persistirlos en un sistema de archivos. Utiliza *Apache Kafka* como intermediario distribuido de mensajería y *Hadoop YARN* para la asignación y programación de recursos distribuidos. *Samza* adopta el servicio del administrador de recursos YARN para proporcionar tolerancia a fallos, aislamiento del procesador, seguridad y administración de recursos en el clúster donde se ejecuta.
- **Apache Storm:** Es una herramienta de código abierto que sirve para procesar datos estructurados y no estructurados en tiempo real. *Storm* es tolerante a fallos, por lo que es adecuado para análisis de datos en tiempo real, aprendizaje automático, computación secuencial e iterativa. Un programa *Storm* es representado por un gráfico acíclico dirigido (DAG).
- **Apache Spark:** Es un potente *framework* de procesamiento fácil de usar con el que se pueden realizar análisis eficientes de datos heterogéneos. Fue desarrollado originalmente en *UC Berkeley* en 2009. *Spark* tiene varias ventajas en comparación con otros marcos de procesamiento como *Hadoop MapReduce* y *Storm*, como la eficiencia, la tolerancia a fallos y la velocidad. Un concepto clave de *Spark* son los conjuntos de datos distribuidos resilientes (RDD)[6]. Un RDD es básicamente una colección inmutable de objetos esparcidos a través de un clúster que ejecute *Spark*. Para procesamiento en tiempo real, *Spark* dispone de dos bibliotecas, *Spark Streaming* con uso de RDDs y *Spark Structured Streaming* con el uso de *dataframes* [7]. Ambas opciones permiten escalabilidad y procesamiento de datos en tiempo real con muy alto rendimiento.
- **Apache Flink:** Es un *framework* de código abierto para procesar datos tanto en tiempo real como por lotes. Proporciona varios beneficios frente a herramientas clásicas, como la tolerancia a fallos y el cálculo a gran escala. El modelo de programación de *Flink* es similar a *MapReduce*. En contraste con *MapReduce*, *Flink* ofrece funciones adicionales de alto nivel como unir, filtrar y agregar. *Flink* permite el procesamiento iterativo y el cálculo en tiempo real de los datos de transmisión

recopilados con diferentes herramientas como *Flume* y *Kafka*. Ofrece varias API en un nivel abstracto, lo que permite al usuario iniciar la computación distribuida de forma transparente y sencilla.

Una vez conocidas las herramientas de procesamiento *Big Data* más populares, se observa que cumplen con la necesidad del problema y que encajarían con los objetivos que tenemos en esta sección del proyecto.

Para conocer cual de las tecnologías debemos escoger es necesario realizar una comparación activa [5]. En primer lugar, aunque la asignación de recursos en *Storm* se garantiza de forma dinámica y transparente, propiedad que gestiona mejor que el resto de tecnologías, ésta solo puede resolver problemas de procesamiento en tiempo real y con funcionalidades limitadas. Además, es relativamente complicada la creación de aplicaciones de *Storm* debido a los recursos limitados de los que dispone.

En el caso de *Samza*, el modelo a bajo nivel de los mensajes basado en *Kafka* que ofrece le otorga una gran flexibilidad, pero esta opción presenta limitaciones en cuanto a la gestión de errores producidos y la optimización de recursos automática. Cuando un nodo intermediario falla, los mensajes ubicados en el sistema de archivos se pierden y no se pueden recuperar [5].

Si estudiamos el uso de *Spark* este destaca en el procesamiento en memoria y el uso de los micro-lotes, especialmente en procesamiento iterativo e incremental. A pesar de que *Spark* es conocido por ser un *framework* de procesamiento muy rápido debido al concepto de RDD que utiliza, posee bajo rendimiento en comparación con otras tecnologías del mismo campo. En cambio, gracias al uso de micro-lotes en su funcionalidad de tiempo real, garantiza la tolerancia a fallos.

*Flink* comparte similitudes y características con *Spark*, también ofrece un buen rendimiento de procesamiento cuando se trata de datos complejos. Ambos destacan en el uso de herramientas gráficas para visualizar el estado y el flujo de las tareas ejecutadas. *Flink* tiene una latencia más baja y una implementación de función de ventana más avanzada, mientras que *Spark* tiene una implementación más madura y un uso más amplio. A la hora de elegir uno de ellos, si realmente es necesario un nivel de latencia de pocos milisegundos y se dispone de una arquitectura sólida, *Flink* es el más idóneo [5]. De lo contrario, se debe escoger *Spark*.

Finalmente, en el proyecto escogemos *Apache Spark*. Debido a los objetivos establecidos y las necesidades tecnológicas mostradas se podría usar *Spark* o *Flink* de forma prácticamente indiferente, ambas resuelven el problema. Aun así, *Spark* tiene mucha más documentación y encaja mejor con el resto de componentes del sistema. Esto se debe a que sus estructuras de datos son mucho más flexibles que las de *Flink*. Por ello, analizando la necesidad del proyecto de disponer de más recursos y capacidades para el desarrollo, en contraposición de obtener una gran velocidad de transmisión, *Spark* encaja mejor con el problema a resolver en el proyecto.

### 3.2.2. Nuestra propuesta

En el proyecto usamos la tecnología de procesamiento *Big Data* llamada *Apache Spark Structured Streaming* para realizar el procesamiento en tiempo real de los datos de consumo cruzados con la ocupación de los edificios para poder realizar un sistema de monitorización completo[8, 6].

*Spark Structured Streaming* es un motor de procesamiento en tiempo real construido como API de *Spark* que se encuentra por encima de *Spark SQL*, permitiendo realizar ingestas y sus posteriores transformaciones. *Apache Spark* es muy conocido en el paradigma *Big Data* ya que fue diseñado para el procesamiento de datos distribuidos de forma rápido y de propósito general. Permite orquestar, distribuir y monitorizar aplicaciones que constan de múltiples tareas de procesamiento de datos sobre una sola maquina o varias máquinas de un clúster [?, 6].

Para comprender cómo funciona *Spark Structured Streaming* es esencial entender el modelo de programación sobre el que se fundamenta esta tecnología. El pilar principal dentro de *Spark Structured Streaming* para el tratamiento de datos en tiempo real es el uso de una tabla agregada continua procesada en memoria principal. A diferencia del procesamiento con *MapReduce* que se realiza sobre memoria secundaria. En *Spark Structured Streaming* existe el concepto de *micro-batching* en función de su intervalo de lote, pero tiene cierta distinción en la transmisión de datos que lo hace más cercano a la transmisión real. En *Spark Structured Streaming* no hay un concepto de lote como tal, los datos recibidos en la aplicación se agregan al flujo de datos que existe continuamente. Cada fila de la secuencia de datos se procesa y el resultado se actualiza en la tabla de entrada con flujo continuo. Cada cierto tiempo establecido (*triggers*) se actualiza la tabla de resultado. Esta forma de actuar da oportunidad de obtener su resultado (actualizado, solo el último resultado o todos los resultados) dependiendo del modo de sus acciones. Normalmente siempre que la tabla de resultados se actualiza queremos plasmar el resultado en otro destino.

Un detalle a tener en cuenta respecto al uso de esta tecnología es el evitar usar los *RDD* (usados en *Spark Streaming*). *Spark Structured Streaming* usa *DataFrames* por defecto para realizar las operaciones de transmisión. Si se compara la latencia y velocidad en las operaciones, la mayoría de resultados conducen a una misma dirección: los *DataFrames* están más optimizados en términos de procesamiento y proporcionan más opciones para agregaciones y otras operaciones con una variedad de funciones disponibles[?].

Por último, cabe destacar que el procesamiento en tiempo real se ha caracterizado históricamente por su problemática con la latencia en la generación y entrega de datos. *Spark Streaming*, otra herramienta, trabaja por lotes, no diferencia si un evento se generó antes del lote actual y, por tanto, pertenecía al lote anterior, generando así una información final menos precisa. Con *Spark Structured Streaming* esto no ocurre, ya que se controla la marca de tiempo del evento comprobando si está incluida en los datos recibidos, gestionando así coherentemente los datos que llegan tarde, como se muestra en la figura 3.1.

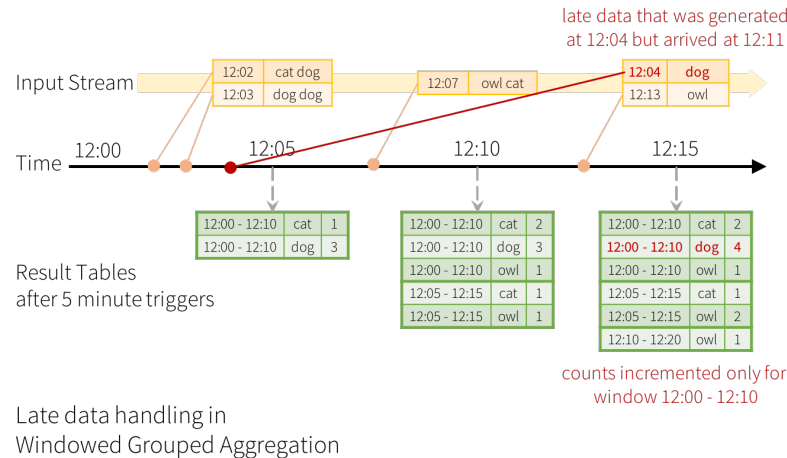


Figura 3.1: Gestión de datos tardíos en Spark Structured Streaming

Sin embargo, para ejecuciones de larga duración, es necesario que el sistema limite la cantidad de estado en memoria que acumula. Si esto no se realizara, la memoria podría llenarse rápidamente. Esto significa que el sistema necesita saber cuándo se puede eliminar un agregado antiguo del estado en memoria porque la aplicación ya no recibirá datos tardíos para ese agregado. Para habilitar esto, se usan las marcas de agua o *watermarking*, que permiten que el motor busque el dato más nuevo y elimine de memoria el más antiguo a tratar según este *watermarking* [4, 6, 7]. Para una ventana específica que finaliza en el momento  $T$ , el motor mantendrá el estado y permitirá que los datos tardíos actualicen el estado hasta:

$$(\text{máximo tiempo de evento visto por el motor} - \text{umbral tardío}) > T$$

### 3.3. Optimización presupuestaria multiobjetivo

Las energías renovables están resultando ser una revolución en el panorama energético, ya que son muy numerosas las ventajas que incluyen. Cumplir con las ya demandadas medidas medioambientales para la sostenibilidad se ha convertido desde hace años en un motivo muy relevante para utilizar energías renovables como fuentes energéticas. Además, la autonomía que otorgan es también muy importante y a tener en cuenta a la hora de plantear la actualización de un sistema energético en marcha o la construcción de uno nuevo.

#### 3.3.1. La Unión Europea demanda eficiencia energética

El Plan de Eficiencia Energética 2011 de la Unión Europea, siendo el último plan de eficiencia energética impulsado por la UE a fecha de publicación de esta memoria, estaba muy dirigido a fomentar el ahorro de energía por medio de la aplicación de soluciones eficientes. Uno de los sectores con mayor potencial era y sigue siendo el sector público, donde los edificios consumen mucha energía finalmente desperdiciada. Uno de los objetivos del plan consistía en instar al sector público a tomar medidas para que comenzara un

proceso de regeneración de los edificios públicos, convirtiendo a éstos en edificios energéticamente eficientes.

El sector energético público representa el 19 % del PIB de la UE, y sus edificios son el 13 % de los existentes en el parque inmobiliario europeo. Por tanto, es esencial redirigir las políticas energéticas de la gestión de los edificios públicos, como los que gestiona la universidad, para poder obtener mejoras en la eficiencia energética.

Además, según este plan, los organismos públicos deberían haber elevado el grado de eficiencia energética del que disponían sus edificios en el 2011. Se deberá actualizar la renovación de sus instalaciones cada cierto tiempo y estarán obligados a renovar al menos el 3 % de sus edificios cada año. Por otra parte, el sector edificación consume el 25 % del total de energía final de la UE. Dentro de este sector, más del 60 % de este consumo tiene su origen en el uso de las calderas. Dadas estas cifras, se hace significativo el hecho de reducir parte del consumo de este sector, ya que con la aplicación de la tecnología adecuada se podría llegar a reducir el 75 % del consumo de un edificio medio.

La renovación de instalaciones que van quedando obsoletas es una de las medidas a realizar, incluyendo en la medida de lo posible las energías renovables, que son mucho menos dañinas para el medioambiente.

La eficiencia energética es de gran importancia para la Unión Europea y, gracias al uso de análisis de datos, podemos producir una solución barata en despliegue que considere energías renovables para que cualquier edificio se pueda adherir a este plan. Consecuentemente, este proyecto busca ajustar correctamente los presupuestos energéticos, teniendo en cuenta los múltiples factores involucrados, para evitar el desperdicio energético y el desgaste medioambiental por parte de la universidad.

### **3.3.2. Presupuestos energéticos multiobjetivo**

Los sistemas energéticos actualmente requieren constante mantenimiento y actualizaciones frecuentes. Dada la urgente y relevante necesidad de diseñar sistemas energéticos que minimicen los desperdicios energéticos, utilicen lo máximo posible las energías renovables y que sean lo más baratos posible, la agilidad en el diseño es esencial. La necesidad de un sistema que cuente con una funcionalidad que permita presupuestar una instalación fotovoltaica óptima, es muy relevante actualmente.

Debido a la necesidad de renovación de la infraestructura energética pública cada poco tiempo y a la necesidad de ajustar la configuración de las instalaciones lo máximo posible para reducir costes, ésta funcionalidad resultaría muy útil para los responsables de una instalación eléctrica.

Para lograr desarrollar esta funcionalidad es necesario poder encontrar una configuración óptima de paneles solares y baterías. Para ello, debemos considerar la intención de cumplir varios propósitos simultáneamente: queremos incluir lo máximo posible la energía fotovoltaica en el sistema pero gastando lo mínimo posible[9, 10, 11].

Este problema se clasifica como un problema de optimización multiobjetivo[12]. Los problemas de optimización se tratan de encontrar una solución que represente el valor óptimo para una función objetivo concreta. La optimización con un único cumple la función  $f : S \rightarrow N$ , donde  $S \subset \mathbb{R}^n$  y  $N \subset \mathbb{R}$ . En el caso de la optimización multiobjetivo, se busca la optimización simultánea de más de un objetivo siguiendo la expresión  $f : S \rightarrow T$ , donde  $S \subset \mathbb{R}^n$  y  $T \subset \mathbb{R}^k$ , lo que podría producir que no exista un elemento de  $S$  que produzca un óptimo de forma simultánea para cada uno de los  $k$  objetivos que componen  $f$ . Esto se deberá a la existencia de conflictos entre objetivos, que harán que la mejora de uno de ellos dé lugar a un empeoramiento de algún otro. Así, el problema tendrá múltiples soluciones óptimas que conforman el *Frente de Pareto*[13]. De esta forma, el problema de optimización multiobjetivo puede formularse como:

$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} && \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_k(\mathbf{x})]^T \\ & \text{subject to} && g_i(\mathbf{x}) \geq 0 \quad i = 1, \dots, m, \\ & && h_i(\mathbf{x}) = 0 \quad i = 1, \dots, p \end{aligned} \tag{3.1}$$

Encontrar el vector  $\mathbf{x}^* = [x_1^*, \dots, x_n^*]^T$  que satisfaga las  $m$  restricciones de desigualdad:

$$g_i(\mathbf{x}^*) \geq 0 \quad i = 1, \dots, m \tag{3.2}$$

Las  $p$  restricciones de igualdad:

$$h_i(\mathbf{x}^*) = 0 \quad i = 1, \dots, p \tag{3.3}$$

Las funciones que describen nuestro problema,  $\mathbf{f}(\mathbf{x})$ ,  $g(\mathbf{x})$  y  $h(\mathbf{x})$ , serán descritos en detalle en la sección 4.3.1.

Las funciones que describen nuestro problema,  $\mathbf{f}(\mathbf{x}) = [C(X_P, X_B), R(X_P, X_B)]$ ,  $g(\mathbf{x}) = [-X_P, -X_B, -X_B C_B - X_P C_P + B]$  y  $h(\mathbf{x}) = []$ , serán descritos en detalle en la sección 4.3.1.

Para obtener una solución óptima, deberán ser tenidos en cuenta parámetros laterales, los cuales son particulares del problema:

- Con respecto a la instalación fotovoltaica, es necesario considerar como parámetros la frecuencia con la que es necesario actualizar los paneles y las baterías del sistema, además de qué coste tienen los componentes a ser actualizados.
- Con respecto a la localización, debemos tener en cuenta la radiación que podría percibir un panel solar en las coordenadas indicadas. Este aspecto debe ser promediado pero, dado el contexto del problema, también se deben considerar los picos de actividad, tanto máximos como mínimos. Esto puede ser balanceado utilizando correctamente las baterías, que también deberán estar configuradas por el algoritmo: cuántas necesitará la instalación.

El algoritmo tiene casos en los que no retornará una configuración como solución. Podría ser posible no poder suplir la cantidad de energía necesaria por un presupuesto muy limitado. En estos casos, el algoritmo lo notificará como resultado. En cualquier otro caso, el algoritmo devolverá una configuración optimizada en número de paneles y número de baterías.

### 3.3.3. Algoritmia evolutiva

Para resolver problemas multiobjetivo, una posibilidad es utilizar algoritmos evolutivos[14]. Esta opción se remonta a finales de los 1960s en que la tesis doctoral de Rosenberg (1967) indicó la posibilidad de usar algoritmos genéticos en este dominio.

Los algoritmos evolutivos que resuelven estos problemas se basan en el concepto de eficiencia de Pareto. Este concepto se basa en, dado un conjunto de elementos candidato a ser solución, iterar de tal forma que, realizando cambios en al menos un elemento, éste mejore y los demás, al menos, no se vean perjudicados. Cuando ya no es posible realizar mejoras sobre ningún elemento del conjunto solución, se considera ésta como solución *pareto-óptima*.

El algoritmo que hemos seleccionado es Evolución Diferencial[15], el cual ha resultado ser el algoritmo que mejores resultados ha proporcionado en la resolución de problemas de esta naturaleza. El algoritmo Evolución Diferencial se caracteriza por ser simple, eficiente y veloz, proporcionando las mejores soluciones.

#### Evolución Diferencial

Evolución Diferencial es una técnica metaheurística basada en poblaciones de vectores numéricos. Entre las ventajas más destacadas de este algoritmo encontramos la simplicidad, eficiencia, propiedad de búsqueda local y velocidad. El proceso seguido por Evolución Diferencial para resolver un problema de optimización se caracteriza por iterar sobre una población de vectores para hacer evolucionar soluciones candidatas con respecto a una función de aptitud llamada *fitness*. El *fitness* evalúa cómo de bien se ajusta la solución candidata a la función objetivo para la cual se quiere optimizar.

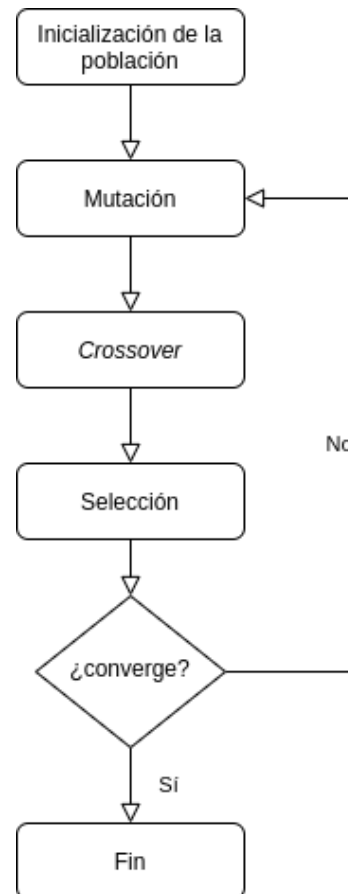


Figura 3.2: Diagrama de flujo del algoritmo Evolución Diferencial

En la figura 3.2 se muestra un diagrama de flujo del proceso de optimización llevado a cabo en el algoritmo Evolución Diferencial. Los pasos que muestra el diagrama son los siguientes[16]:

1. **Inicialización de la población:** el algoritmo comienza generando de manera aleatoria una población inicial de  $N$  vectores que son soluciones potenciales para el problema de optimización que se resuelve.
2. **Mutación:** para cada vector objetivo en cada generación, se crea un vector mutado, que se obtiene mediante perturbaciones de vectores que conforman la población actual.
3. **Crossover:** este proceso tiene como propósito agregar diversidad genética a la población. Para ello, el operador de *crossover* genera vectores de prueba utilizando información que copia de dos vectores diferentes. En particular, Evolución Diferencial cruza cada vector objetivo con su correspondiente vector mutado dependiendo de la probabilidad de recombinación de la población.
4. **Selección:** para conformar la siguiente generación, cada vector objetivo es comparado con su correspondiente vector de prueba, usando una función de *fitness*  $f(x)$ . El vector con el mejor valor de *fitness* es el que integrará la población de la siguiente generación. Para evaluar la configuración generada, necesitamos definir un valor de *fitness* para cada individuo. La función de *fitness*  $f$  que asigna un valor de aptitud



a cada vector objetivo. En nuestro problema esta función está representada por la ecuación 4.7.

Las restricciones en Evolución Diferencial son tratadas mediante una estrategia de penalización. En consecuencia, éstas adoptan un valor de *fitness* más alto de acuerdo al grado de violación de cada restricción.

5. **Condición de parada:** la nueva generación atraviesa el proceso descrito desde la mutación hasta la selección. El ciclo se repite hasta que se satisface la condición de parada, que consiste en alcanzar un máximo número de generaciones ( $g_{max}$ ).

### 3.3.4. Posibles soluciones

La complejidad de un problema de estas características limita las opciones existentes para paliar la necesidad de presupuestar sistemas energéticos que consideren energías renovables. Aun así, el mercado actualmente ofrece las siguientes opciones [17] que, parcialmente se ajustan a la problemática de este proyecto:

- **Aurora:** Permite diseñar a través de modelos en 3D una instalación fotovoltaica. La herramienta simula las condiciones de la localización en la cual se quiere realizar la instalación. Su uso requiere superar una curva de aprendizaje poco amigable y utilizar un ordenador con muy buenas prestaciones. Además, toma mucho tiempo realizar presupuestos ya que es necesario maquetar toda la instalación componente a componente. Es software privativo y de alto coste.
- **BlueSol:** Proporciona una experiencia de uso muy similar a la herramienta anterior, pero su interfaz de usuario es más sencilla. También es software privativo y de alto coste.
- **SAM:** Permite modelar sistemas energéticos utilizando diferentes tipos de energías renovables. Se trata de software privativo diseñado por el Departamento de Energía de los Estados Unidos. A pesar de ser muy completa, carece de funcionalidades concretas del diseño de instalaciones fotovoltaicas, como la consideración de parámetros laterales.

Estas aplicaciones tienen todas en común que son costosas, tienen una interfaz de usuario insuficientemente amigable con el usuario y, lo más importante, ninguna de ellas es software libre.

### 3.3.5. Nuestra propuesta

Valorando las opciones analizadas en la sección anterior, debemos reconocer que tienen gran precisión y flexibilidad, pero como indicábamos anteriormente, ninguna de ellas es software libre; todas estas opciones tienen una licencia privativa que restringe la posibilidad de editar las funcionalidades que aportan. Esto imposibilita modificar el código del software que utilizan para poder adaptar el funcionamiento del algoritmo utilizado por detrás a una situación particular. Nuestra propuesta creada es de software libre, por lo que cualquier interesado podrá modificar el código, compartirlo y estudiarlo.

El desarrollo de una herramienta de estas características bajo una licencia libre facilita a los clientes de este mercado la entrada a éste, consiguiendo así que más instituciones,

empresas y particulares puedan sumarse al cambio de la energía fotovoltaica. Además, al ser software libre, el desarrollo de esta funcionalidad de la plataforma es colaborativo, por lo que puede ser mejorado por los usuarios que lo utilicen. Así, conseguimos proporcionarle a cualquier interesado en consumir energía limpia una plataforma abierta con la cual presupuestar y diseñar una instalación fotovoltaica.

Otro motivo muy importante por el cual consideramos necesario el desarrollo de esta funcionalidad considerando el mercado actual, es que ninguna de estas herramientas permite adherir la instalación fotovoltaica que diseñas a una instalación energética ya existente o con energía de otras fuentes, sea fotovoltaica o no. Nuestra herramienta incluye la posibilidad de evaluar el diseño de la instalación fotovoltaica contra un sistema ya existente y decidiendo si se quiere utilizar otras fuentes energéticas o no.

Nuestra propuesta integra en la plataforma una solución a este problema utilizando código *Python* con la librería *jMetalPy*. Consideramos otras librerías, como *pymoo* o *platypus*, pero ninguna de las dos se ajustaba a las necesidades del problema como *jMetalPy*. Además, ésta es la opción más liviana y con más documentación.

# Capítulo 4

## Requisitos

Hasta ahora, hemos discutido varias problemáticas pertinentes a la situación actual del mercado de herramientas software de gestión energética, señalando qué consideramos que debe ser realizado para resolverlas. Para lograr reducir la incertidumbre y la complejidad en el desarrollo, en este capítulo explicamos qué servicios debe ofrecer nuestra solución para que concuerde con lo propuesto anteriormente.

### 4.1. Plataforma web como base de un Energy Management System

Una de las características principales del proyecto es ofrecer las funcionalidades energéticas necesarias en un único sistema unificado, base principal del EMS. Este proyecto implementa las siguientes funcionalidades pertenecientes a un EMS:

1. **Monitorización en tiempo real:** Permite detectar incidencias en cualquier momento, sin esperar a que se consuma un intervalo de tiempo.
2. **Análisis de datos históricos:** Muestra información a través de analizar datos pasados con el objetivo de descubrir tendencias y patrones.
3. **Informes personalizados:** Presenta información con las necesidades de visualización de datos pertinentes al conjunto de los usuarios de la plataforma.

Esta plataforma debe ser simple e intuitiva, mostrando información concreta y que cualquier usuario, técnico o no, pueda usarla sin problema y sin necesidad de guías de usuario. Para que esto se cumpla es necesario generar una curva de aprendizaje pequeña.

La plataforma tiene que ser consistente en su totalidad, tanto de forma visual manteniendo el mismo estilo en todas las secciones como en la estandarización de la simbología. También es importante que el usuario reciba *feedback* directo, mostrando de forma intuitiva cuándo un elemento ha dejado de funcionar o ha surgido algún error. El usuario debe saber, en todo momento y de forma sencilla, qué está sucediendo en la plataforma.

En un EMS se ofrecen muchos tipos de información y secciones diferentes que en la mayoría de las ocasiones no es relevante para todos los usuarios y lo único que pueden

provocar es una sobrecarga de información. Por ello, el proyecto tendrá perfiles o roles diferenciados dentro de la plataforma, donde cada uno tendrá acceso a la información que necesita.

Con estas limitaciones conseguimos que un perfil acceda únicamente a la información que necesita para concentrar mejor su análisis. Además, de esta forma reforzamos más la seguridad de la información a la que accede cada usuario.

Los diferentes perfiles y la información a la cual pueden acceder son los siguientes:

- **Responsable de suministros y personal de mantenimiento:** Accede a los informes gráficos agregados de datos de la universidad cruzados con estimaciones de energía fotovoltaica y al sistema de monitorización en tiempo real que compara los consumos frente a la ocupación.
- **Gestor energético:** Puede acceder a la información específica analítica de optimización multiobjetivo de configuración de equipo fotovoltaico. También tiene acceso al sistema de monitorización en tiempo real que compara los consumos frente a la ocupación.
- **Estudiante y personal docente:** Acceden únicamente a los informes gráficos agregados de datos de la universidad cruzados con estimaciones de energía fotovoltaica.
- **Administrador del sistema:** Tiene acceso completo a todas las secciones de la plataforma y la sección de administración de usuarios y grupos.
- **Usuario no registrado:** Puede acceder a información básica de las instalaciones pero no puede acceder a ningún análisis. Para registrarse, puede solicitar acceso a la plataforma. Esta solicitud deberá ser aceptada o denegada por algún administrador del sistema.

## 4.2. Representación de la información

El proyecto, al tratarse de un *Energy Management System* centrado en generar conocimiento analítico para los responsables de las instalaciones energéticas, debe mostrar la información de forma que sea sencilla de entender por los perfiles que la consuman. Además, debe ser lo suficientemente completa para poder extraer valor de su estudio. Antes de especificar los requisitos del análisis, es necesario conocer qué parámetros se pueden especificar en el caso a estudiar. Para esto, se provee de formularios previos al análisis donde se dispone de todas los posibles parámetros a introducir. La plataforma debe proveer de un total de tres formularios, uno para cada sección analítica del proyecto.

### 4.2.1. Datos de entrada de informes

El formulario web de recogida de datos para la generación de informes que estudian los datos de consumo energético de un determinado centro ofrece, primero, la posibilidad de selección del centro y del año para el cual se quieren realizar los análisis. Una vez

rellenadas estas dos opciones, se muestran los parámetros de secciones específicas. Si solo se completa el centro y el año, se mostrará un análisis base sobre los datos de origen. El resto de selecciones deben ser opcionales y concretas de cada uno de los análisis diferentes. Las secciones son las siguientes:

- **KPIs:** Datos generales del estudio de los datos en su conjunto y sus metadatos. Esta sección muestra métricas desde un aspecto más amplio, que pueden ayudar al usuario a obtener un pensamiento más global de a qué información se enfrenta en el estudio. Se especifican con valores discretos.
- **Comparación entre energía activa y reactiva:** Esta sección muestra una gráfica que compara los niveles de energía activa usada. Los compara con la energía reactiva usada y muestra también el máximo total para estudiar junto a las otras dos unidades.
- **Consumo con uso de energía fotovoltaica:** Encontramos varias gráficas que muestran el consumo estimado si la instalación seleccionada usase una configuración de placas solares definida en el formulario. Las gráficas se muestran de diferentes formas, descrito en métricas energéticas y monetarias. Además, se ofrece la posibilidad de mostrar la gráfica en periodos descritos por horas, meses o incluso ambas opciones.
- **Uso de baterías:** Con una configuración detallada de baterías, se muestra, junto con el uso de energía fotovoltaica, la diferencia entre usar o no usar baterías para el almacenaje de energía en nuestro sistema y el ahorro que esto conllevaría.

En el caso de la monitorización de consumo frente a la ocupación de un edificio, la web debe ofrecer un formulario sencillo que permite seleccionar entre los edificios disponibles y los parámetros de configuración de las alarmas. Estos parámetros son el porcentaje de consumo que puede superar como máximo comparándolo con la ocupación y el consumo por persona en kWh. La web debe proporcionar valores por defecto para estos parámetros.

Por último, la plataforma debe disponer de un formulario específico para la optimización multiobjetivo, la cual nos muestra cuál es la mejor configuración de energía solar, consumo y baterías para un centro en particular. Para esta solicitud el usuario debe encontrar, como con el resto de los formularios, un desplegable que nos permite seleccionar el centro de donde se obtendrán las coordenadas para el análisis, campos para introducir las restricciones de las placas solares como inclinación u orientación y los límites presupuestarios de los que se dispone para la construcción o renovación del sistema. En este caso no es necesario introducir la fecha de análisis, ya que se realiza el estudio con el año natural de la solicitud[18].

#### 4.2.2. Visualización gráfica

Dada la naturaleza del contexto en el que se desarrolla el proyecto, la visualización de datos es esencial, por lo que es necesario acotar qué datos queremos explotar en cada informe y de qué manera: qué clase de gráficas y qué consideraciones queremos utilizar. Para ello definimos a través del uso de bocetos y explicaciones, qué datos utiliza cada informe y qué parámetros acepta.

Se quiere disponer de dos gráficas principales para mostrar la información solicitada en el formulario. La más específica es una gráfica de rangos temporales, donde la ventana temporal seleccionada se encuentra en el eje de coordenadas 'x' y las métricas específicas en el eje 'y'. Es necesario, debido a la elevada cantidad de datos que procesa el sistema, que cuando el usuario seleccione con el cursor un elemento gráfico, se visualicen los datos referentes a esa selección con detalle. También es necesario que inicialmente no se muestre todo el conjunto de datos, si no el rango más reciente. Esto es debido a que los datos más recientes suelen ser los más interesantes en un análisis de estas características. Es necesario proporcionar una selección de fechas fluida y configurable por el usuario según quiera y proveer de rangos de fechas por defecto para simplificar su uso. La figura 4.1 muestra un ejemplo de este tipo de gráfica.

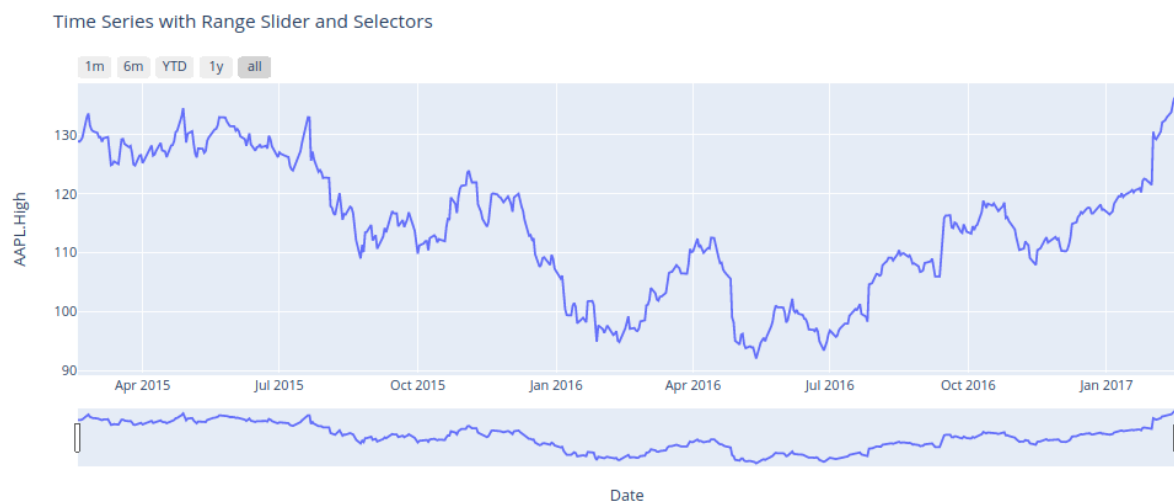


Figura 4.1: Ejemplo de gráfica con ventanas temporales interactivas

El segundo tipo de gráfica está enfocada principalmente a comparar dos o más tipos de métricas con rangos temporales discretos. La gráfica de barras temporal es la que más se ajusta a estos análisis. Esto se debe a los casos en los que se requiere comparar series temporales definidas en pocos valores, como por ejemplo, en análisis por meses. El formato, visualización y configuración de estas gráficas son construidas específicamente para cada solicitud de información diferente. Por tanto, aunque se deba mantener una misma base gráfica, a cada elemento se le añaden todos los detalles que benefician a su análisis final concreto.

Aunque el uso de las gráficas antes nombradas predomina a lo largo de los informes, existen otros elementos a usar para acompañar a estos elementos visuales y que son igual de importantes. El uso de tablas de datos e indicadores específicos, ambos enfocados a mostrar información directa y general sobre los datos a partir de técnicas de estadística aplicada que ayudan al entendimiento de la información de un informe. En definitiva, se deben mostrar metadatos y valores fijos analíticos sobre el conjunto de los datos, no sobre el enunciado del caso de uso en particular.

### 4.3. Optimización multiobjetivo

En el caso del cálculo de configuración fotovoltaica óptima, será necesario introducir

los datos del formulario correspondiente. Tras esto, el sistema mostrará, tras realizar el análisis y los cálculos pertinentes, una propuesta de configuración de elementos a colocar en la instalación con la que obtendremos el máximo beneficio.

En este resultado del análisis, obtendremos además datos técnicos del sistema, como consumo por red eléctrica estimado. La configuración recomendada de energía fotovoltaica muestra la potencia necesaria, el número de paneles en paralelo y la energía que provee cada módulo a utilizar. Además, la configuración indica qué características debe tener el regulador de carga usado y cuáles son las características de las baterías a usar, estimando detalles como días de autonomía, capacidades y número de baterías necesarias en serie. Por último, la configuración muestra la configuración que debe tener el inversor principal.

Este servicio muestra también los costes totales de la distribución ofrecida, desglosada por componentes. De esta forma, el técnico puede modificar algún componente si bajo su criterio considera que no es el adecuado. Si se ha estipulado un límite de coste para el presupuesto, el sistema deberá intentar ofrecer el óptimo dentro considerando el máximo. Si este valor es menor que el mínimo posible a presupuestar y, por tanto, no hay ninguna posibilidad en el rango dado, el sistema mostrará el presupuesto más bajo posible, aunque supere el presupuesto indicado, mostrando una notificación si esto se ha producido.

### 4.3.1. Formulación del problema

El objetivo del problema es calcular la cantidad de paneles solares  $X_P$  y baterías  $X_B$ , dada una cierta demanda energética y perfil de generación solar. La variable de decisión es por tanto:

$$\mathbf{x} = [X_P, X_B] \quad (4.1)$$

Los paneles solares están definidos por su cantidad  $X_P$  en  $\text{m}^2$ , su perfil de generación efectiva  $g(t)$  en  $\text{kW}/\text{m}^2$ , función de la ubicación y eficiencia de los paneles, su precio inicial  $C_P$  en  $\text{€}/\text{m}^2$ , y su vida útil  $V_P$  en días. El perfil de generación solar tiene en cuenta las pérdidas por una inclinación no perpendicular con el sol y la eficiencia del panel. En la práctica, el perfil de generación es medido a intervalos regulares y es usado como una serie temporal  $\mathbf{g} = [g_1, \dots, g_N]$ , donde  $N$  es el número de datos por día, 24 por defecto. La generación en  $\text{kW}$  de los paneles solares queda como  $X_P g_i, i = 1, \dots, N$ .

Las baterías se definen de una forma similar, cantidad  $X_B$  en  $\text{kWh}$ , su precio inicial  $C_B$  en  $\text{€}/\text{kWh}$ , y su vida útil  $V_B$  en días. La carga de las baterías se define usando la serie temporal  $\mathbf{q} = [q_1, \dots, q_N]$ , donde la carga se incrementa cuando hay exceso de generación solar, y disminuye cuando las baterías son usadas para cubrir la demanda. Esta carga está limitada por cero en su límite inferior y por  $X_B$  en su límite superior.

La demanda energética es definida por la serie temporal  $\mathbf{d} = [d_1, \dots, d_N]$ , en  $\text{kWh}$ . Por comodidad, definimos la demanda media la demanda total  $d_T = \sum_{i=1}^N d_i$  y la generación total  $g_T = \sum_{i=1}^N g_i$ .

Para poder plantear el problema de una forma sencilla, hemos considerado las siguientes suposiciones:

- La capacidad de las baterías es constante durante el periodo del análisis. En la práctica, la capacidad de las baterías disminuye con su uso, pero por simplicidad utilizamos una capacidad efectiva a lo largo de su vida útil.
- La producción de los paneles solares es constante durante el periodo del análisis.

- La red eléctrica tiene capacidad ilimitada de producir y aceptar electricidad. Para las cantidades usadas en este problema esto no supone un problema.
- El precio de compra y venta a la red es constante. Aunque es posible contratar una tarifa con precio constante, la práctica más común es precio dinámico que varía a lo largo del día, pero no lo incluimos en este análisis.
- Tanto las baterías como los paneles solares pueden ser comprados en cualquier cantidad, y el precio por unidad es independiente de la cantidad. Esto simplifica el problema de optimización ya que nos permite usar variables reales en vez de un número finito de opciones.
- Las baterías pueden cargarse y descargarse de forma instantánea. Esto simplifica el problema y permite ignorar los efectos dinámicos de los picos de demanda o caída de generación solar. En este trabajo no supone un problema porque los datos de demanda tienen poca resolución temporal.

En la resolución del problema deben utilizarse varias funciones objetivo para calcular el número óptimo de paneles y baterías. El primer objetivo a considerar es el coste diario de la instalación. Este se puede dividir en coste fijo  $C_f$ , y coste variable  $C_t$ . El coste fijo viene dado por el coste de los paneles solares y las baterías, teniendo en cuenta su vida útil. Queda expresado de la siguiente forma:

$$C_f = X_B \frac{C_B}{V_B} + X_P \frac{C_P}{V_P} \quad (4.2)$$

El coste variable viene dado por el coste de la electricidad, que puede ser negativo en caso de vender a la red eléctrica. Para ello definimos un precio de compra  $C_e$  y un precio de venta  $C_v$  en €/kWh.

En cada instante de tiempo  $i$ , la demanda  $d_i$  es principalmente satisfecha por los paneles solares. Si los paneles solares no producen suficiente energía, la demanda restante es satisfecha por las baterías. Si las baterías están vacías, entonces se compra electricidad de la red para satisfacer la demanda. Por el contrario, si los paneles solares producen suficiente energía para cubrir la demanda, el excedente de energía se usa para cargar las baterías. Si las baterías estuvieran llenas, entonces el excedente de generación de electricidad se vende a la red. El diagrama 4.2 muestra este proceso de uso del sistema. Usando este modelo, el coste total se puede expresar de la siguiente forma:

$$C = C_f + C_t = X_B \frac{C_B}{V_B} + X_P \frac{C_P}{V_P} + \sum_{i=0}^N \Delta C_i(X_B, X_P, \mathbf{g}, \mathbf{d}) \quad (4.3)$$

donde  $\Delta C_i$  varía en función de si la demanda es satisfecha por la red eléctrica con coste  $C_e$  por kWh, o el excedente de generación es vendido con coste  $-C_v$  por kWh.

Una vez calculado el coste diario  $C$ , se puede calcular el tiempo  $T$  que tarda la instalación en ser rentable. Para ello, el beneficio tiene que ser igual al coste inicial  $C_I$ . Esto queda expresado por:

$$CT = C_I = X_B C_B + X_P C_P, \quad (4.4)$$

$$T = \frac{X_B C_B + X_P C_P}{C}. \quad (4.5)$$



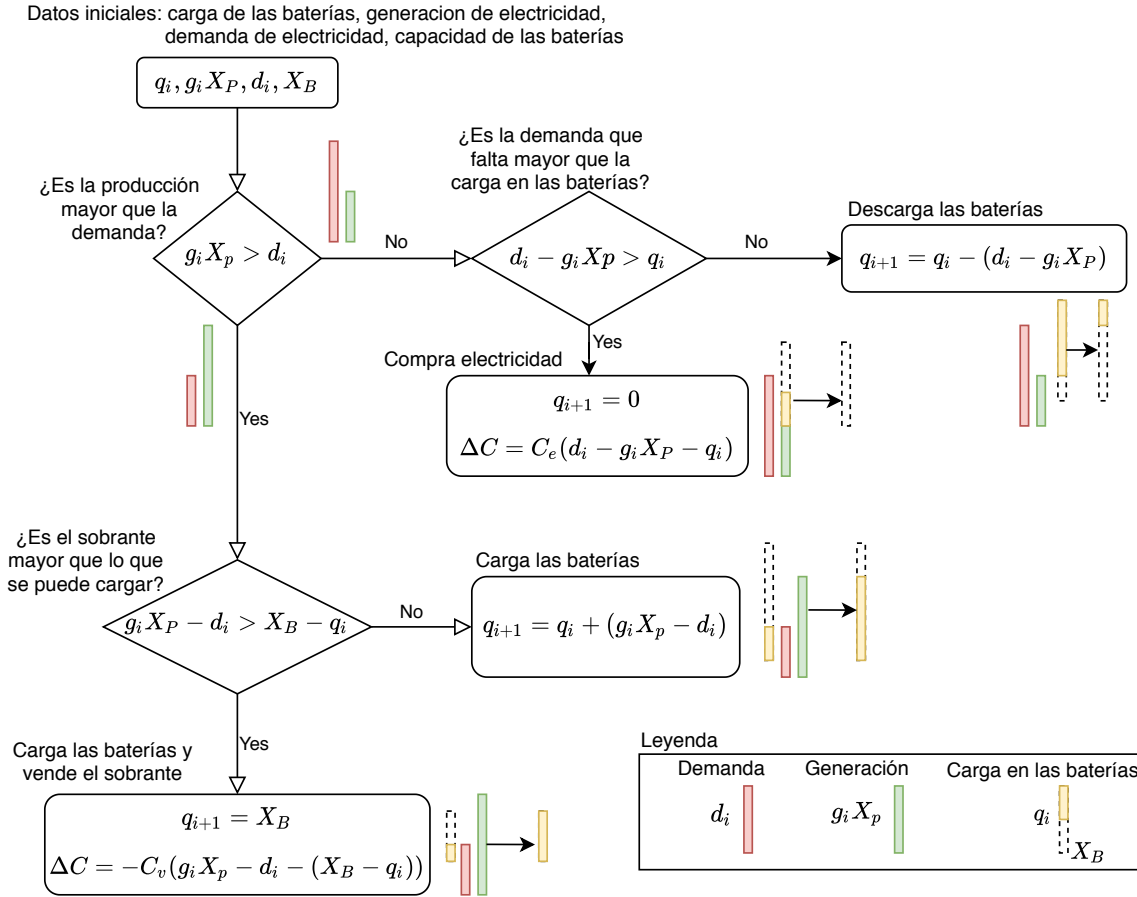


Figura 4.2: Diagrama de satisfacción de la demanda energética usando paneles solares, baterías y la red eléctrica. Este árbol de decisión se utiliza en cada instante temporal.

Otro criterio para dimensionalizar la instalación es la producción de  $CO_2$  por la actividad energética. Los paneles solares y las baterías no generan  $CO_2$ , pero el consumo por red eléctrica sí. Por tanto, el segundo objetivo es minimizar el consumo por la red eléctrica, representado con la serie temporal  $\mathbf{r} = [r_0, \dots, r_N]$ . Este consumo solo se da si la generación de los paneles solares y la carga en las baterías es menor que la demanda de electricidad. En esos casos, el consumo de la red eléctrica para el instante  $i$  es:

$$r_i = d_i - g_i X_P - q_i \quad \text{si } g_i X_P + q_i < d_i \quad (4.6)$$

El consumo de la red eléctrica total  $R$  es la suma de los consumos horarios.

Queda, por tanto, la siguiente función multiobjetivo:

$$\mathbf{f}(\mathbf{x}) = [C(X_P, X_B), R(X_P, X_B)] \quad (4.7)$$

Las restricciones  $g(\mathbf{x})$  a aplicar en la resolución del problema son las siguientes:

- El número de paneles es positivo:

$$X_P \geq 0 \quad (4.8)$$

- La capacidad de las baterías es positiva:

$$X_B \geq 0 \quad (4.9)$$

- El coste inicial es menor que el presupuesto  $B$ :

$$X_B C_B + X_P C_P \leq B \quad (4.10)$$

En este problema de optimización no hay funciones de igualdad  $h(\mathbf{x})$ . El enunciado del problema, compuesto por todo lo mencionado hasta ahora, se formula en la siguiente expresión:

$$\begin{aligned} & \underset{X_B, X_P}{\text{minimize}} && \left[ X_B \frac{C_B}{V_B} + X_P \frac{C_P}{V_P} + \sum_{i=0}^N \Delta C_i(X_B, X_P, \mathbf{g}, \mathbf{d}), \sum_{i=0}^N r_i(X_B, X_P, \mathbf{g}, \mathbf{d}) \right] \\ & \text{subject to} && X_P \geq 0, \\ & && X_B \geq 0, \\ & && X_B C_B + X_P C_P \leq B \end{aligned} \quad (4.11)$$

Además de la solución óptima, incluimos los siguientes métodos para comparar y entender el problema. Estos métodos son ilustrativos de la forma en la cual muchas instalaciones son dimensionalizadas:

1. **Solución sin paneles solares ni baterías.** En este caso,  $X_P = 0, X_B = 0$  y el coste es  $C_e d_T$
2. **Solución de sistema aislado.** En este caso toda la demanda viene de los paneles solares,  $\sum_{i=1}^N d_i = \sum_{i=1}^N g_i X_P$ .

$$X_P^a = \frac{\sum_{i=1}^N d_i}{\sum_{i=1}^N g_i} = \frac{d_T}{g_T}. \quad (4.12)$$

El tamaño de las baterías viendo dado por las restricciones de cargar toda la energía solar cuando no hay demanda y satisfacer la demanda cuando no hay energía solar. Para un instante de tiempo  $i$ , la carga evoluciona de forma

$$q_{i+1} = q_i + X_P g_i - d_i. \quad (4.13)$$

Esta expresión se aplica secuencialmente en cada instante temporal partiendo de un nivel de carga cero. Conocido el nivel de carga, la capacidad de las baterías es entonces la amplitud máxima de carga, expresado por:

$$X_B^a = \max \mathbf{q} - \min \mathbf{q} \quad (4.14)$$

y el nivel de carga inicial es:

$$q_0 = -\min \mathbf{q}. \quad (4.15)$$

Se puede ver dos casos extremos, *a*) la generación y la demanda son proporcionales,  $d_i \propto g_i$ . En este caso no se necesitan baterías, *b*) la generación y la demanda nunca coinciden y una precede la otra. En este caso la capacidad necesaria sería la demanda total:  $X_B = d_T$ . El coste de esta solución es:

$$C^a = (\max \mathbf{q} - \min \mathbf{q}) \frac{C_B}{V_B} + \frac{d_T}{g_T} \frac{C_P}{V_P} \quad (4.16)$$

Este coste puede ser muy superior al coste óptimo si el coste de las baterías o los paneles solares es muy alto.

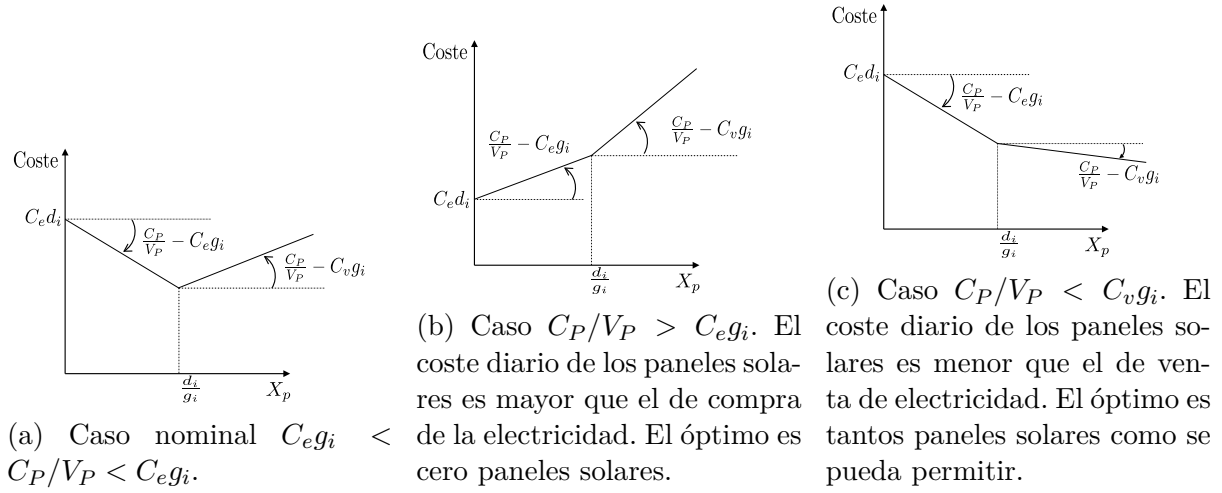


Figura 4.3: Análisis del coste horario en función del número de paneles solares en el caso sin baterías. Los arcos con flecha indican la pendiente de la recta correspondiente.

3. **Solución sin baterías:** Todo el excedente se vende a la red y toda la demanda no cubierta por los paneles solares se compra de la red. El coste es

$$C = X_P \frac{C_P}{V_P} + \sum_{i=1}^N [C_e \max(0, d_i - X_P g_i) - C_v \max(0, X_P g_i - d_i)] \quad (4.17)$$

Esta expresión es difícil de evaluar. La presencia de la función máximo hace que no sea posible calcular el valor óptimo directamente. Para entender mejor su comportamiento, incluimos el coste fijo dentro de la suma:

$$C = \sum_{i=1}^N [C_e \max(0, d_i - X_P g_i) - C_v \max(0, X_P g_i - d_i) + X_P \frac{C_P}{NV_P}] \quad (4.18)$$

Esto nos permite analizar el coste en función del número de paneles solares, para cada instante del cual se dispongan datos de demanda y generación solar. La figura 4.3 muestra la forma de la función para un cierto instante de tiempo  $i$ . El coste para esa hora en función de los paneles solares tiene la siguiente forma: si no hay paneles solares el coste viene por comprar toda la electricidad

$$C_i(X_P = 0) = C_e d_i \quad (4.19)$$

Cada panel solar tiene un coste fijo  $X_P C_P/V_P$ . Además, los paneles solares evitan comprar la electricidad, generando un coste  $-C_e g_i X_P$ . Cuando el numero de paneles solares genera tanta electricidad como la demanda,  $d_i = g_i X_P$ , añadir paneles solares solo permite vender el excedente de electricidad a un coste  $-C_v g_i X_P$ . Por tanto, la pendiente de  $C(X_P)$  es  $C_P/V_P - C_e g_i$  para un numero de paneles solares inferior al critico, y  $C_P/V_P - C_v g_i$  para un numero de paneles solares superior al critico.

En función del valor de  $C_P/V_P$ ,  $C_e$ ,  $C_v$ ,  $d_i$ , y  $g_i$ , es posible que el óptimo instantáneo sea cero, infinito, o un número finito de paneles distinto de cero. Para calcular el

óptimo para todos los instantes de tiempo, primero calculamos el número crítico de paneles solares.

$$X_i^c = \frac{d_i}{g_i}, \quad i = 1, \dots, N \quad (4.20)$$

Definimos la nueva serie temporal  $\mathbf{g}^C$  con los mismos datos que  $\mathbf{g}$ , pero ordenada de menor a mayor utilizando  $\mathbf{X}^c$  como criterio de ordenación. Con esta nueva serie, el coste diario tiene una forma similar a la figura 4.4. Cada vez que se cruza un número de paneles solares crítico, la pendiente se incrementa al cambiar de comprar electricidad a  $C_e$ , a venderla por  $C_v$ , siempre y cuando  $C_e > C_v$ . Por comodidad definimos el ratio de compra y venta  $\eta = C_v/C_e$ . En caso contrario ( $C_e < C_v$ ), los usuarios podrían obtener beneficio comprando la electricidad y revendiéndola después. El óptimo se da cuando la pendiente se vuelve positiva:

$$\frac{C_P}{V_P} - \sum_{i=k}^N C_e g_i - \sum_{i=1}^{k-1} C_v g_i > 0 \quad (4.21)$$

Expresado de otra forma:

$$\sum_{i=k}^N C_e g_i + \sum_{i=1}^{k-1} C_v g_i < \frac{C_P}{V_P} \quad (4.22)$$

El proceso para encontrar el número óptimo de paneles es el siguiente: se evalúa la ecuación 4.22 con valores de  $k$  más altos hasta que se encuentra un valor  $k^*$  que hace que se cumpla la ecuación. Finalmente, el número de paneles óptimo es  $X_{k^*}^c$ .

Hay dos casos particulares en los cuales se puede calcular la solución de forma sencilla:

- a) Si el coste fijo es más alto que el coste de la electricidad,  $C_e g_T < \frac{C_P}{V_P}$ , la ecuación 4.22 se satisface inmediatamente y la solución son cero paneles.
- b) Si el coste fijo es más barato que el coste de vender la electricidad,  $C_v g_T > \frac{C_P}{V_P}$ , en este caso la ecuación 4.22 nunca se satisface, y el óptimo son tantos paneles como sean posibles.

Para expresar de forma compacta la condición en la que no hay un número trivial de paneles solares, definimos la siguiente variable:

$$\xi_P = \frac{C_P}{V_P g_T C_e} \quad (4.23)$$

Utilizando esta nueva variable, el número de paneles no es cero o infinito si  $\eta < \xi_P < 1$ , donde  $\eta$  es el ratio de precios de compra y venta:  $\eta = C_v/C_e$ . La figura 4.9a muestra este criterio en un caso de ejemplo.

Una vez vistas las soluciones particulares, podemos analizar en problema general y calcular la solución directamente en ciertos casos. Por ejemplo, la capacidad de las baterías tiene que ser menor que la capacidad de generación de los paneles,  $X_B < X_p g_T$ . De lo contrario, la capacidad extra nunca sería cargada.

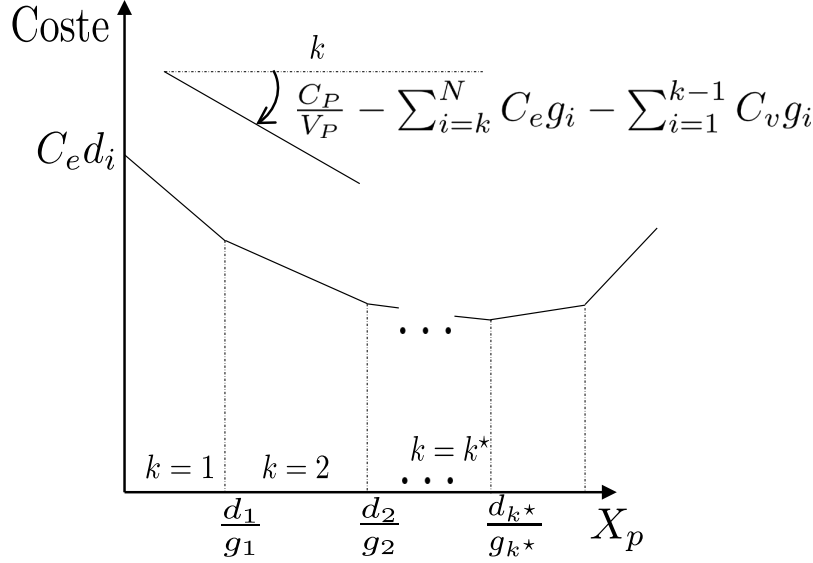


Figura 4.4: Coste diario en función del número de paneles. El coste está compuesto por hasta  $N$  rectas y cambia de pendiente en cada número crítico de paneles. El arco con flecha indica la pendiente de la recta en un intervalo genérico  $k$ .

Otra condición para usar baterías es que el coste diario por kWh tiene que ser menor que lo que nos hubiera costado comprar esa misma cantidad de electricidad. Para verlo mejor, pongámonos en el siguiente caso: hay una generación  $X_P g_i$  en el instante  $i$  con una demanda  $d_i$  menor, por tanto hay un excedente instantáneo de energía  $X_P g_i - d_i$ . Llamemos a este excedente  $s$ . Existen dos posibilidades a realizar con el excedente: 1) venderlo, generando un coste  $-C_v s$ , 2) guardarlo en la batería, generando un coste diario de la batería  $s C_B / V_B$  y después evitar un coste de  $C_e s$  al no tener que comprar la electricidad. Para que el uso de baterías sea rentable, el coste tiene que ser menor comparado con el caso de no tenerla:

$$s \frac{C_B}{V_B} - C_e s < -C_v s \quad (4.24)$$

$$\frac{C_B}{V_B} < C_e - C_v = C_e(1 - \eta) \quad (4.25)$$

Expresado de una forma más compacta:

$$\xi_B = \frac{C_B}{V_B C_e(1 - \eta)} < 1 \quad (4.26)$$

La figura 4.9b muestra como este valor se cumple para un caso de ejemplo. Además, la capacidad de las baterías nunca va ser mayor que el del caso aislado:

$$0 \leq X_B \leq X_B^a \leq d_T \quad (4.27)$$

### 4.3.2. Ejemplos de optimización

Para ilustrar la resolución del problema de optimización energética, mostramos aquí los resultados usando datos creados a propósito para este ejemplo. La demanda energética y la irradiación se muestran en la Figura 4.5 y representan perfiles típicos. Los parámetros para

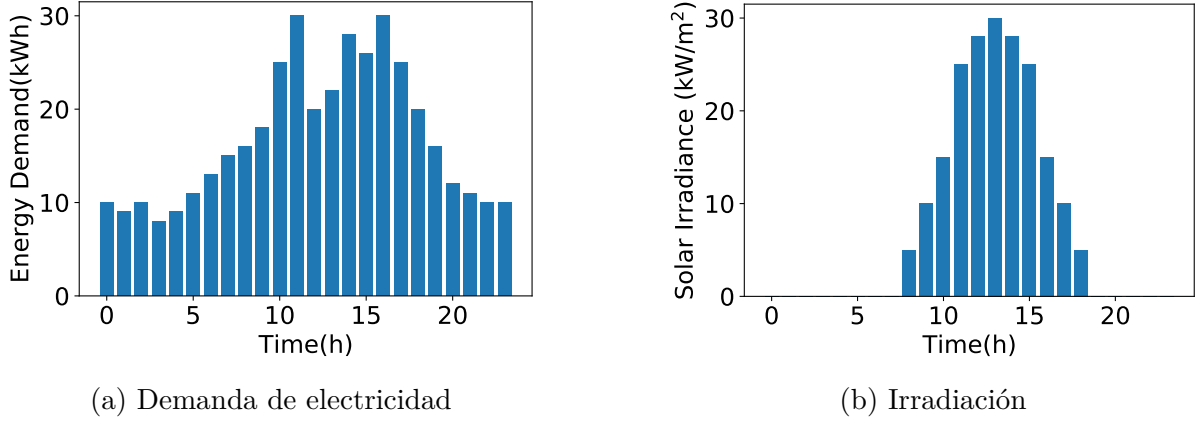


Figura 4.5: Series temporales para el ejemplo

este ejemplo son:  $C_e = 1$ ,  $C_v = 0,5$ ,  $C_b = 0,25$ ,  $V_b = 1$ ,  $C_p = 170$ ,  $V_p = 1$ . Estos valores no tienen significado físico y han sido elegidos para dar una solución óptima interesante. Con estos valores se obtiene  $d_T = 404$ ,  $g_T = 196$ ,  $\eta = 0,5$ ,  $\xi_P = 0,86$ ,  $\xi_B = 0,5$ . El valor de  $\xi_P$  se encuentra dentro del rango  $[\eta, 1]$  por tanto la solución óptima va a ser finita y distinto de cero. De forma similar,  $\xi_B < 1$  y el número óptimo de baterías va a ser mayor que cero.

La solución de sistema aislado es  $X_P = 2,06$ ,  $X_B = 163,7$ , y la solución óptima de coste, utilizando el algoritmo evolutivo NSGAI, es  $X_P = 1,07$ ,  $X_B = 23,37$ . Ambas soluciones se muestran en la figura 4.6, junto a series temporales de diferentes configuraciones típicas. Se puede observar que la solución sin baterías ni paneles solares compra toda la electricidad de la red, mientras que la solución solo con paneles solares vende todo el excedente a la red. La solución de sistema aislado ni compra ni vende a la red, y la solución óptima es una mezcla de todas las soluciones anteriores.

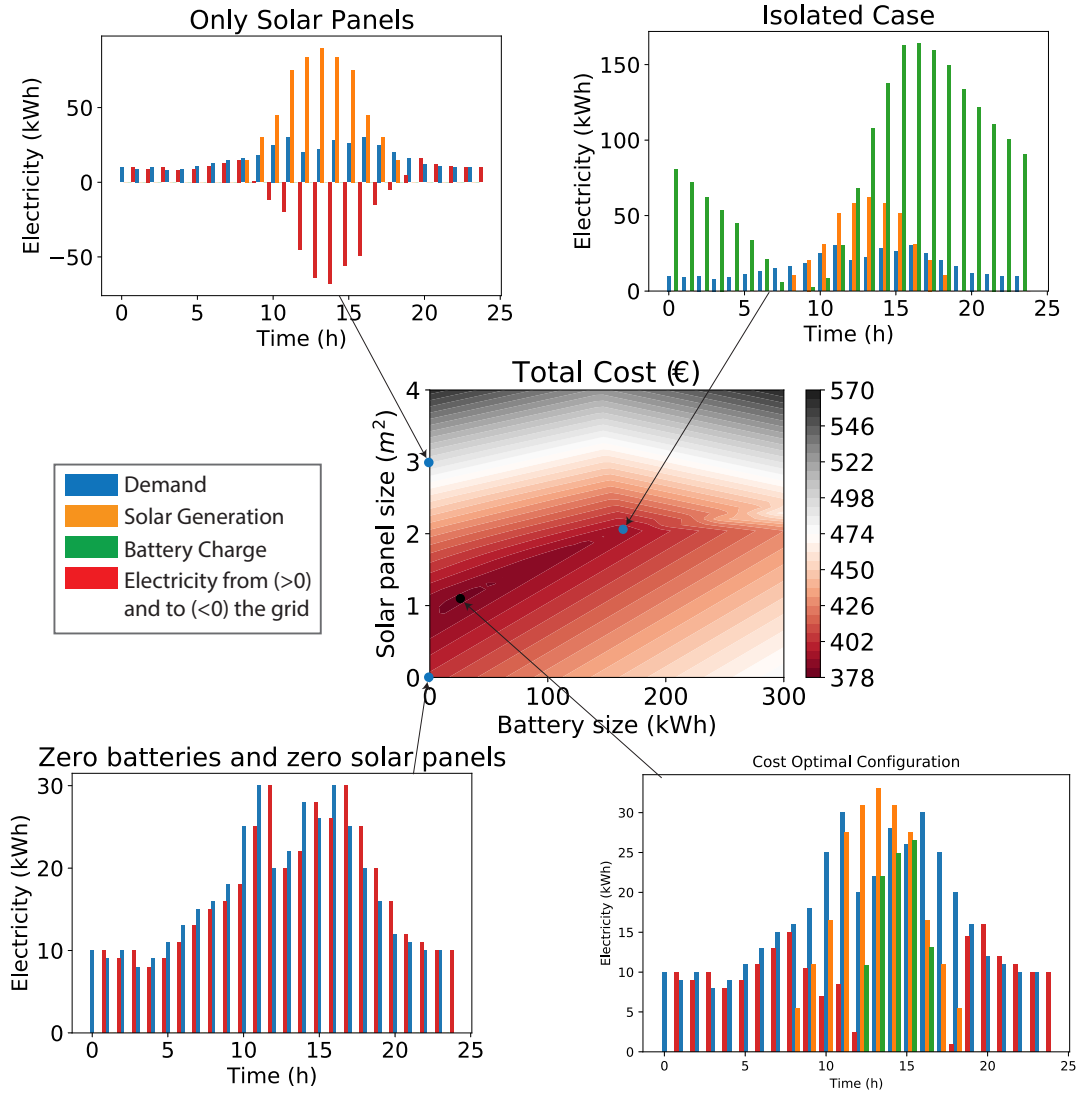


Figura 4.6: Ejemplo de optimización del número de paneles y baterías. La figura central muestra un diagrama de contorno del coste para cada configuración de paneles y baterías. En cada esquina, se muestran configuraciones particulares con flechas apuntando a la configuración que representan: solo paneles solares, que vende toda la energía sobrante a la red; sin baterías ni paneles, que compra toda la energía; el caso aislado, que ni se compra ni se vende de la red; y el caso óptimo, que es una mezcla de los anteriores.

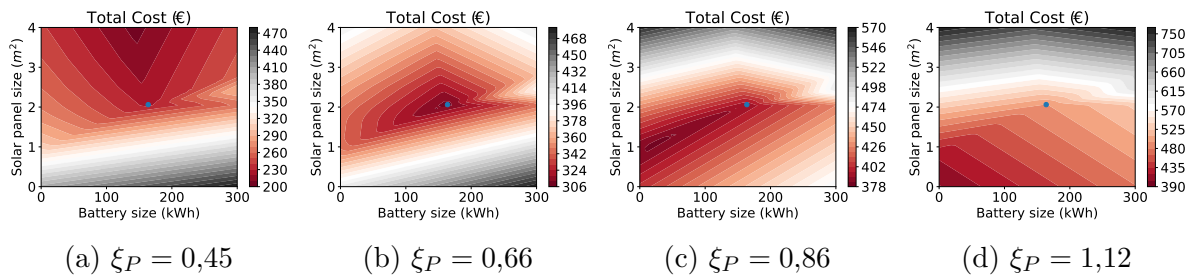


Figura 4.7: Coste total para diferentes valores de  $\xi_P = \frac{C_P}{V_{PgT}C_e}$ . Todas las gráficas tienen  $\xi_B = 0,5$ ,  $\eta = 0,5$

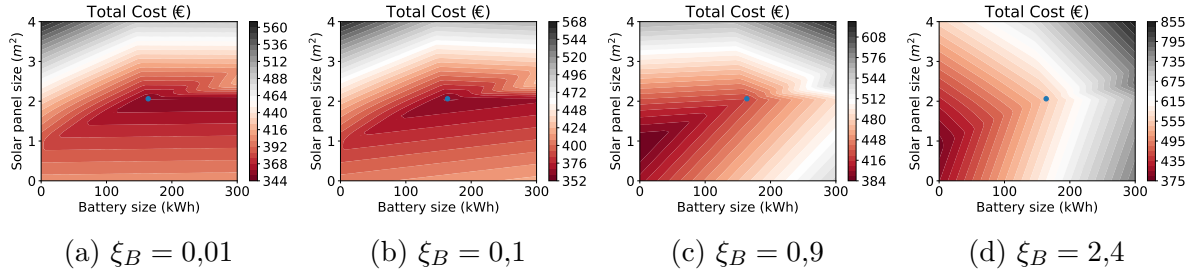


Figura 4.8: Coste total para diferentes valores de  $\xi_B = \frac{C_B}{V_B C_e (1-\eta)}$ . Todas las gráficas tienen  $\xi_P = 0,76, \eta = 0,5$

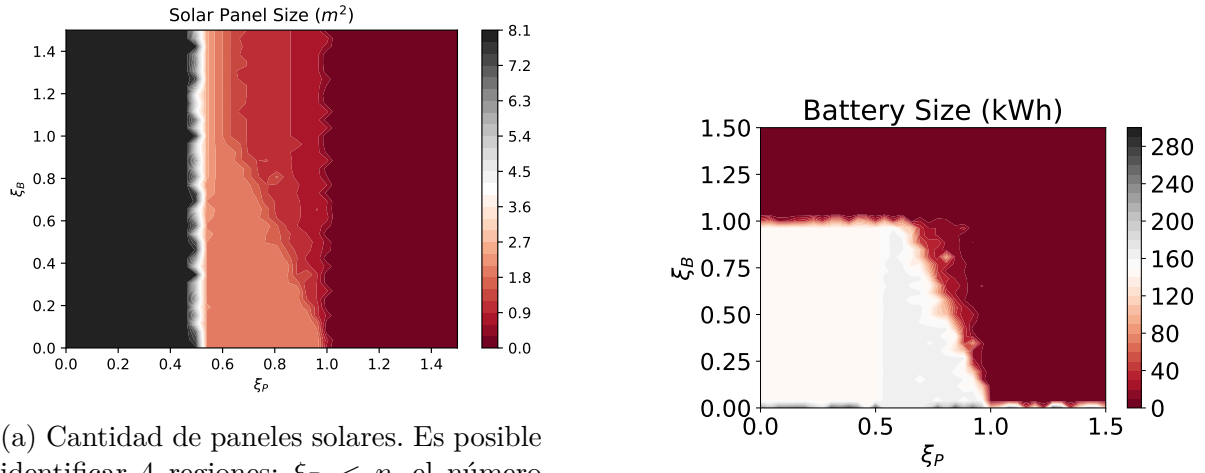
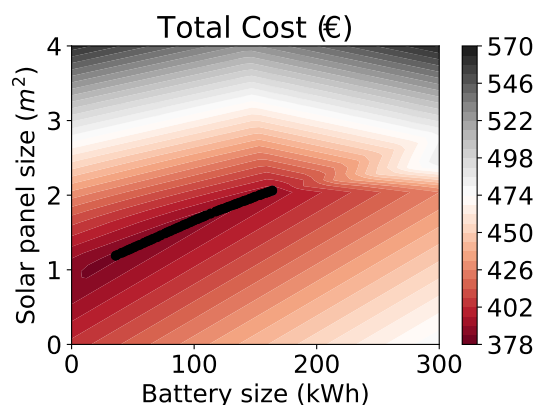
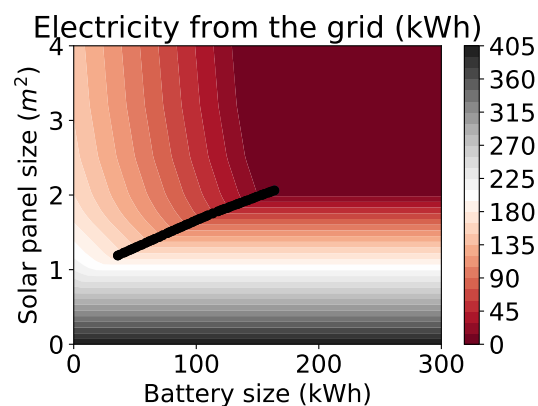


Figura 4.9: Número de paneles solares y de baterías en función del coste diario normalizado





(a) Solución óptima de Pareto en la función objetivo coste diario



(b) Solución óptima de Pareto en la función objetivo cantidad de electricidad de la red eléctrica.

Figura 4.10: Funciones objetivos con el conjunto óptimo de Pareto. Se puede ver como conectan los valores óptimos considerando cada objetivo por separado.

Una vez que también consideramos ambos objetivos, el coste diario y la compra de electricidad de la red eléctrica, la solución no es única sino una serie de soluciones, llamada conjunto óptimo de Pareto. En este ejemplo podemos ver ese conjunto en la figura 4.10.

## 4.4. Monitorización del consumo frente a la ocupación de un edificio

La monitorización del proyecto se corresponde con un sistema de alarmas que muestren gráficas e indicadores con datos agregados. Las alarmas se tratan de avisos visuales a través de la plataforma web. Estos avisos se presentan como una ventana emergente roja que indica qué elemento ha provocado la incidencia y por qué motivo. El principal objetivo de este sistema es proporcionarle al responsable de la instalación en cuestión el estado de ésta en tiempo real.

Para cumplir con los objetivos propuestos, el sistema dispondrá de tres grupos de análisis en esta sección, compuesto por varios elementos y que muestran de forma específica diferentes métricas:

- **Gráficas de visualización directa:** Se deberán mostrar dos gráficas en tiempo real con los datos sin transformar, para ver en las últimas horas cómo ha cambiado tanto el consumo como la ocupación. Estas gráficas son actualizadas dinámicamente cada vez que se reciben datos en la aplicación en tiempo real desde los sensores.
- **Indicador porcentual como sistema de aviso:** Se trata de un valor que muestra el porcentaje de sobre-consumo que se da en un centro. Además, provee de notificaciones en la propia plataforma para avisar si se supera un valor especificado previamente. Es actualizado cada vez que la aplicación recibe datos.
- **Indicadores de consumo y comparativas:** Un conjunto de valores discretos que mostrarán el consumo total acumulado en kWh, la suma de personas que han

pasado por el centro de estudio y el consumo por persona en kWh, valor para el momento actual y desde el inicio de ese día. Además, se mostrará, para el mismo momento, la tendencia con respecto al día anterior. Es actualizado cada vez que la aplicación recibe datos.

#### 4.4.1. Respuesta eficiente en tiempo real

Como cualquier sistema de este tipo, es necesario que la monitorización se realice en tiempo real, de esta forma se consigue explotar todo el potencial del análisis y se puede actuar cuanto antes ante una posible incidencia en la instalación energética.

Un técnico de la instalación, que conoce en todo momento el estado del centro que gestiona, obtiene una ventaja analítica clave. Ésta hace que la toma de decisiones se realice en un espacio de tiempo corto y siempre menor a la toma de decisiones sobre datos históricos. Este detalle debe ser tenido en cuenta al basar las decisiones sobre la configuración de la instalación sobre datos del pasado. Esto conllevaría la pérdida de eficiencia energética y, consecuentemente, sufrir consumos que se podrían haber evitado.

La monitorización del consumo energético no es suficiente para conocer si el nivel de consumo es el esperado. En el caso de un EMS, la información esencial para realizar este análisis es la ocupación en tiempo real de los centros a analizar. Cruzando estos datos junto a los datos de consumo se obtienen resultados mucho más precisos y eficaces. Además, es relevante destacar que un sistema de monitorización de consumo energético debidamente unido a otras métricas es capaz de permitir entender el impacto ambiental de cada edificio y de cada parte de la actividad de una organización. De esta forma, es posible la creación de indicadores de desempeño energético de fácil seguimiento, muy útil para los estándares *ISO 50001* que discutimos anteriormente.

El proyecto está enfocado a cubrir principalmente las necesidades de análisis energético de la Universidad Complutense de Madrid. Uno de los objetivos del proyecto es poder desplegar el sistema en su infraestructura. Esto es un factor relevante por las dimensiones de la información que el sistema debe procesar. El tamaño de datos a procesar simultáneamente en tiempo real es muy elevado, debido a la cantidad de edificios de los que dispone esta institución. Por ello, es necesario desarrollar un sistema que sea capaz de soportar una gran carga de procesamiento simultáneo. Además, que no pierda información si sucede algún problema y que los procese de forma distribuida para que la latencia de las transformaciones sea la menor posible. El *pipeline* necesario para el procesamiento de información agregada en tiempo real requiere de una gestión compleja del dato a lo largo de todo su ciclo de vida, contando con cada uno de los estados que puede tener y las transformaciones que se le quiera realizar.

# Capítulo 5

## Arquitectura software

La plataforma desarrollada en este proyecto posee una arquitectura software que le permite gestionar todas las posibles peticiones de los usuarios de la misma. Las principales características sobre las cuales nos hemos concentrado en el desarrollo han sido la usabilidad, la eficiencia, la portabilidad, la mantenibilidad y la integridad. Los estándares que hemos considerado para cada una de estos atributos de calidad son los siguientes, basándonos en el ISO/IEC 9126:

- **Usabilidad:** La aplicación web sobre la que se presentan las funcionalidades implementadas debe ser sencilla de usar; su uso debe ser accesible para perfiles no técnicos que no conozcan absolutamente nada de cómo ha sido construida la aplicación. El esfuerzo requerido para aprender y operar la aplicación debe ser bajo. El enfoque que hemos utilizado para lograr esto es acercar aquello con lo que puede interactuar el usuario, es decir, la interfaz de usuario, lo máximo posible al dominio del perfil para el cual desarrollamos la aplicación. Para lograr esto hemos dividido el código de la aplicación a través de una serie de abstracciones, las cuales explicaremos más adelante.
- **Eficiencia:** El tiempo de respuesta de la aplicación a las peticiones de los diferentes usuarios debe ser asumible considerando el contexto en el que se encuentra el sistema, es decir, cargas y procesamientos de datos a través de una interfaz web. Considerando las cantidades de datos que trata la aplicación, hemos determinado que ninguna consulta ni operación sobre el sistema debe tardar más de 10 segundos en retornar los resultados.
- **Portabilidad:** Debe ser posible desplegar el software desarrollado con facilidad en sistemas no necesariamente iguales en condiciones a los utilizados durante el desarrollo. Nuestro objetivo ha sido poder portar el software a cualquier sistema que utilice *Linux* como sistema operativo. Para lograr esto hemos desarrollado la aplicación bajo el paradigma basado en contenedores software utilizando la herramienta *Docker*.
- **Mantenibilidad:** El software desarrollado debe poder ser fácilmente cambiado sin necesidad de que los cambios a realizar afecten a funcionalidades que se deseen mantener intactas. Durante el diseño de la arquitectura focalizamos este aspecto en la separación clara entre responsabilidades de componentes, de tal forma que todas las piezas del software posean en su código alta cohesividad en las funcionalidades

de las que son responsables y bajo acoplamiento con funcionalidades de las que no son responsables.

- **Integridad:** El proyecto considera en sus especificaciones diferentes perfiles que tienen acceso a la aplicación pero con permisos diferentes, por lo que no todos los usuarios pueden realizar las mismas consultas de datos. Hemos resuelto este aspecto incluyendo en el diseño un módulo de gestión de usuarios y roles integrado en la propia aplicación; al menos un usuario recibe el rol de administrador con el que puede gestionar a qué contenidos pueden acceder el resto de usuarios. Para realizar la distinción de usuarios éstos deberán iniciar sesión autenticándose en el sistema.

La arquitectura de la aplicación se soporta principalmente sobre el popular *framework* web *Django* para *Python*. En los siguientes puntos detallaremos cómo hemos diseñado y estructurado la arquitectura de nuestro proyecto:

## 5.1. Plataforma web

Las interconexiones e interacciones entre módulos del sistema e interfaces API públicas son muy relevantes en nuestra plataforma, ya que son esenciales en la funcionalidad de ésta. La importancia arquitectónica de estas interacciones es muy grande en el sistema, por lo que el diseño software de la plataforma ha estado ligado en gran medida a los requisitos de estas comunicaciones entre componentes, además del flujo de datos en éstas. En esta sección explicamos cuáles son los orígenes de los datos, cómo han sido diseñados los componentes del sistema y cómo interactúan entre ellos.

### 5.1.1. Componentes del sistema

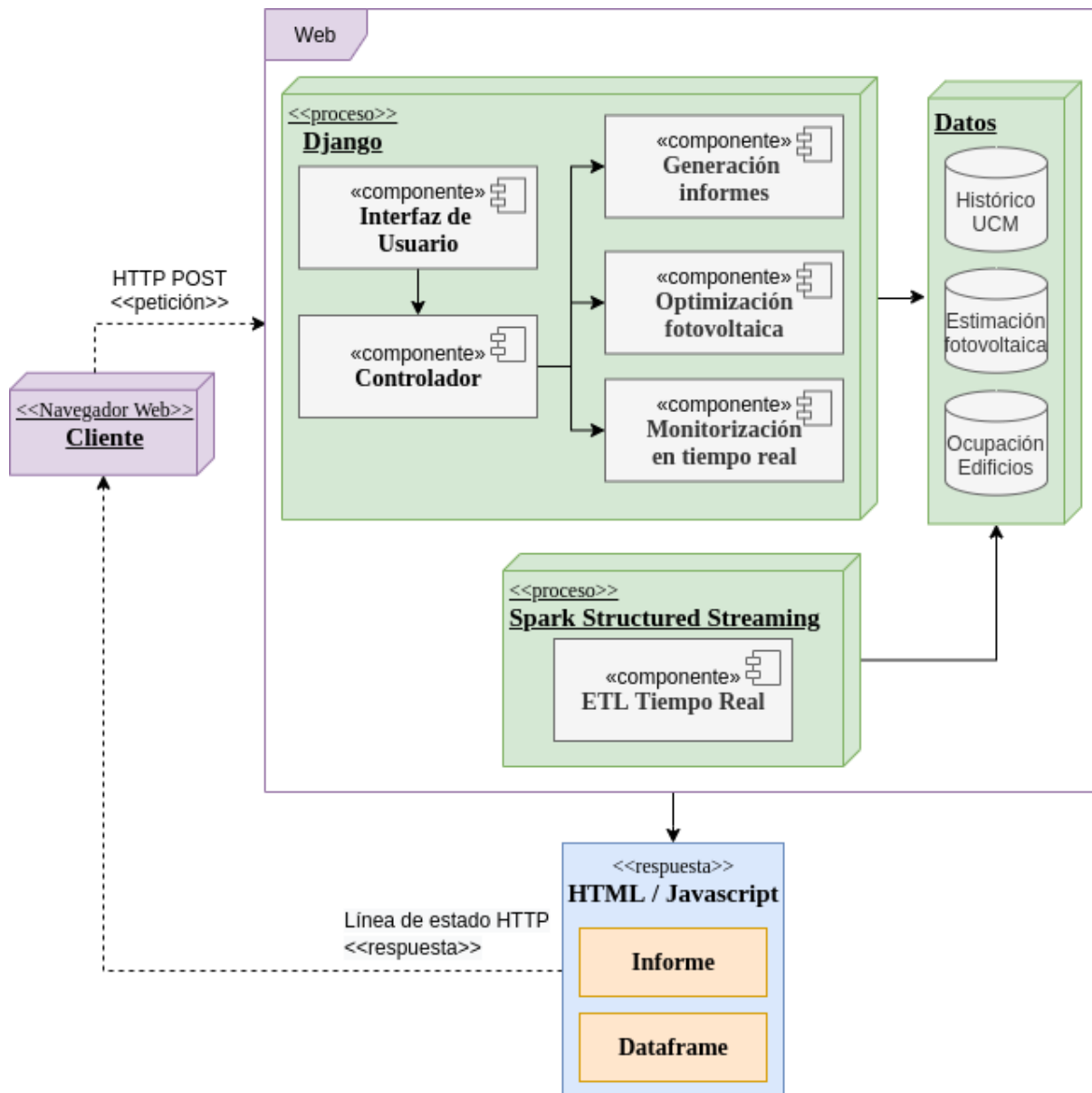


Figura 5.1: Componentes del sistema y sus interacciones tras una petición

El diagrama de la figura 5.1 muestra las interacciones entre componentes en la aplicación desde una petición del cliente del servidor web hasta la generación de la respuesta y la entrega al usuario de los resultados en un documento *HTML*. En este proceso, recogemos una petición POST, la pasamos por un controlador software que la marca asignándole un componente destino que sabe cómo resolverla, éste la recoge, genera un documento *HTML* de respuesta, lo empaqueta en una respuesta y lo envía de vuelta al cliente[2].

Para la construcción y diseño de las gráficas que componen los informes que presentamos a los usuarios, cumpliendo con los requisitos establecidos, utilizamos la librería *Plotly* en *Python*. En el cálculo de la optimización presupuestaria, utilizamos la librería *jMetalPy*[19]. La monitorización de consumo por ocupación es resuelta a través del uso de la librería *PySpark*. Por último, la librería *Beautiful Soup* la usamos para construir el

documento *HTML* que posteriormente le devolvemos al usuario.

Los componentes software que resuelven las peticiones son los siguientes:

- **Vistas:** Construimos todas las vistas a través del diseño de módulos independientes. La cabecera, el pie de página, los menús y todos aquellos elementos comunes a varias vistas, son documentos *HTML* independientes que son importados en cada vista. Estas vistas, por lo tanto, importan los componentes necesarios e implementan aquello pertinente para su funcionalidad. De esta forma, hemos construido un sistema de vistas fácilmente mantenible, extensible, con componentes desacoplados y livianos.

La construcción de estos documentos *HTML* se realiza en código *Python*. Un componente en esta capa recibe las peticiones de construcción de vistas y las redirige a aquellos que conocen cómo construirlas. Cada gráfica dispone de un componente que la construye. Además, estos componentes delegan en otros para elementos comunes como la cabecera o el pie de página.

- **Controlador:** Con el objetivo de construir un sistema MVC (Modelo - Vista - Controlador), desarrollamos un controlador que recoge peticiones desde la interfaz de usuario y las redirecciona a los componentes capaces de resolverlas. Esto es llevado a cabo asignándole un componente destino a cada petición dinámicamente en base a su URL. Esta redirección es resuelta por capas. El componente Controlador contiene múltiples controladores que, construyendo una estructura arbórea, redirige las peticiones a aquellos componentes que conocen de su capa o a los responsables de otras capas. Por lo tanto, si la resolución de una petición le corresponde resolverla a un componente de la segunda capa, ésta pasará por el controlador raíz, éste analizará la URL y decidirá si le corresponde a una capa más profunda o a su capa. Si le corresponde a su capa, le envía la petición al componente responsable y éste la resuelve. Si no, le envía la petición al controlador correspondiente de la siguiente capa y éste realizará el mismo proceso.

La disposición de las capas se corresponde con las funcionalidades anidadas que, en nuestro caso, tienen una profundidad máxima de 2. Cada una de las principales funcionalidades supone una rama nueva en el árbol de posibles redirecciones de la aplicación.

- **Lógica de negocio:** en nuestro proyecto las operaciones y las peticiones son independientes entre ellas, por lo que en nuestro diseño utilizamos lógica de negocio idempotente que no utiliza estado. Así, sustituimos el modelo por una estructura *dataframe* de la librería *pandas* que es independiente a cada petición. Los algoritmos de visualización de datos, así como la optimización multiobjetivo y la monitorización en tiempo real se implementan en módulos de lógica de negocio. De esta forma, los componentes que son responsables de la resolución de las peticiones, que se esconden tras el controlador, tienen el siguiente comportamiento:
  - **Generación de informes:** el componente responsable de esta funcionalidad recibe por parámetros las características del informe a generar. Responde con un documento *HTML* que contiene todas las gráficas solicitadas.
  - **Optimización fotovoltaica:** el componente responsable de esta funcionalidad recibe por parámetros las características de la optimización a realizar.

Responde con un documento *HTML* que contiene por texto la configuración óptima calculada.

- **Monitorización en tiempo real:** el componente responsable de esta funcionalidad recibe por parámetros las características de la monitorización a realizar. Responde con un documento *HTML* que contiene las directrices para construir las gráficas, una ruta a un fichero *CSV* con los datos de la respuesta actualizados en tiempo real y un documento *Javascript* capaz de actualizar las gráficas *HTML* en tiempo real.

### 5.1.2. Especificación del comportamiento

En el proyecto movemos grandes cantidades de datos entre diferentes módulos de la aplicación. Para ello, necesitamos una estructura de datos potente y escalable. Cumpliendo con estos requisitos, decidimos utilizar la estructura de datos *dataframe* de la librería *pandas* ya que además soporta un gran número de operaciones nativas. En la aplicación no mantenemos un estado global para ninguna estructura de datos, por lo que cada petición está enlazada a un conjunto de estructuras *dataframe* que son transportadas por las diferentes capas de la aplicación. Estas estructuras no pueden ser modificadas por instancias de componentes pertinentes a otras peticiones.

Las tres funcionalidades principales del sistema, la visualización de datos, la optimización multiobjetivo presupuestaria y la monitorización de consumo por ocupación, son tratadas de igual manera. Todas las peticiones reciben el mismo tratamiento de redireccionamiento a los componentes responsables de tratarlas.

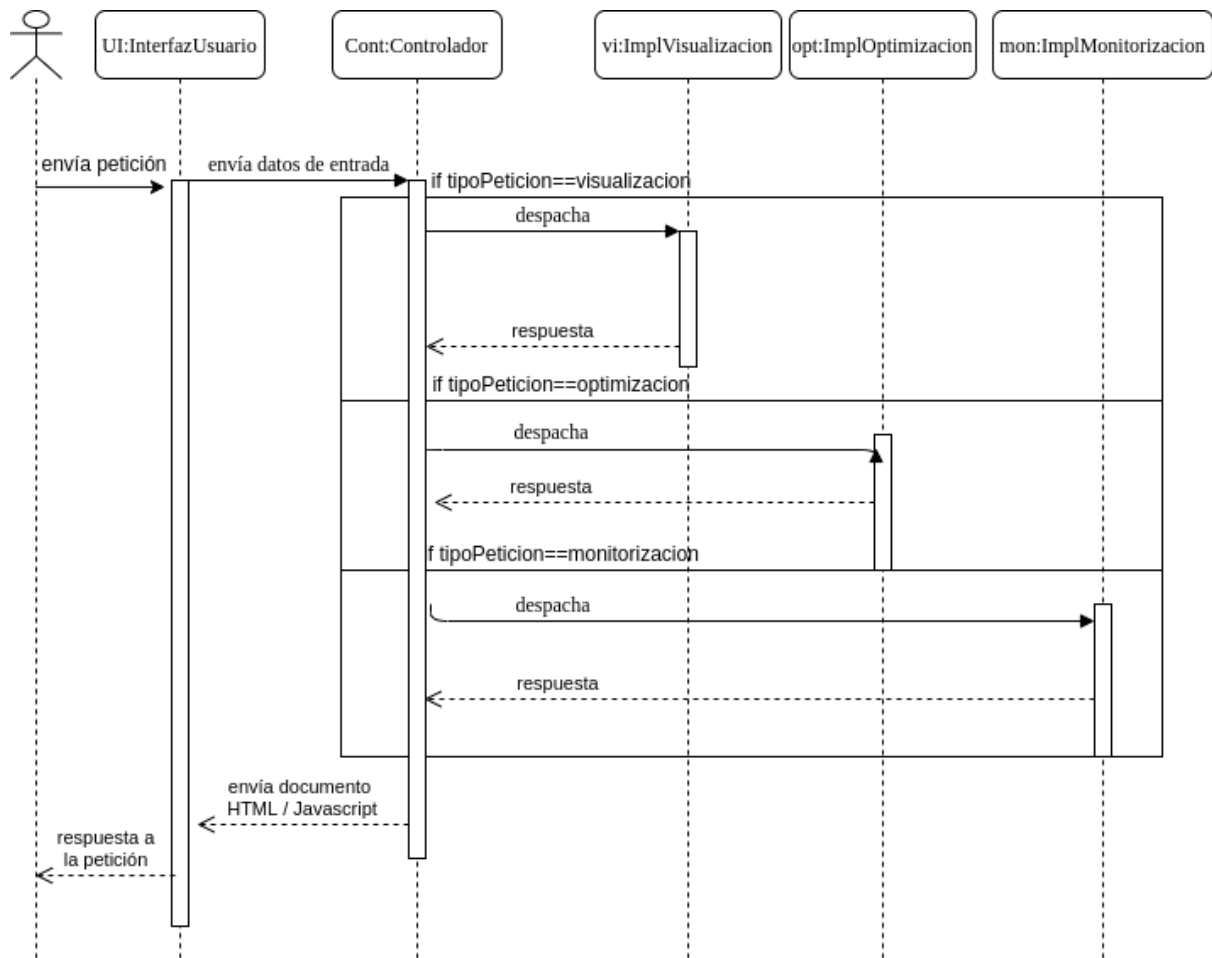


Figura 5.2: Diagrama de secuencia de interacción entre componentes

En el diagrama de la figura 5.2 se muestra con detalle el comportamiento en las interacciones entre los componentes previamente explicados.

## 5.2. Fuentes de datos

Nuestro proyecto está enfocado en la aplicación de herramientas analíticas sobre datos energéticos de la Universidad Complutense de Madrid, por lo que la selección de datos está basada en adaptar la aplicación a los formatos que utiliza la universidad.

### 5.2.1. Datos históricos de consumo de la Universidad Complutense de Madrid

Para el desarrollo de este proyecto, nos han sido proporcionados datos históricos de consumo de cada instalación energética de la Universidad Complutense de Madrid. Estos datos representan información histórica separada en periodos temporales que han sido extraídos directamente de los sensores de las instalaciones. Estos datos no han recibido ninguna clase de preprocesamiento o limpieza antes de ser incluidos en la aplicación.

Los conjuntos de datos de la universidad son ficheros *CSV* que, dado un año natural, contienen cerca de las 10.000 mediciones realizadas en ese año por los sensores instalados



en los sistemas energéticos. La estructura de datos de estos ficheros contiene los siguientes campos:

- **Centro:** Nombre del centro en el cual se han capturado los datos.
- **ID:** Número de identificación del sensor.
- **Fecha:** Día y hora en el cual se capturaron los datos.
- **Consumo:** Consumo percibido por el sensor.
- **Reactiva:** Indicador de energía reactiva.
- **CO<sub>2</sub>:** Nivel de CO<sub>2</sub> percibido por el sensor.
- **PotenciaMax:** Potencia máxima percibida por el sensor.
- **PotenciaAct:** Potencia activa utilizada en el momento en el cual se capturaron los datos.
- **Tarifa:** Intensidad contratada en Amperios.
- **Coste:** Coste contratado del consumo al mes. Cambia cada mes.

Los ficheros han sido almacenados en el repositorio del proyecto y cargados en el mismo sistema. Estos datos han sido utilizados en todas las funcionalidades de la aplicación. El sistema cuenta con un archivo de configuración de tipo *YAML* que actúa como registro caché que indica con qué ficheros (de qué años) cuenta la aplicación y donde se encuentran situados cada una de las fuentes de datos. También estos ficheros actúan como archivos de configuración para la plataforma si es necesario, en particular para los valores que son globales de alguna función y es necesario que pueda ser localizado su valor y su cambio se haga de forma rápida.

### 5.2.2. Estimación de radiación fotovoltaica con web scraping

Para la obtención de los datos referentes a la energía fotovoltaica no se dispone de un acceso directo a ello como en el caso de los datos de las instalaciones, por tanto, se usó como referencia la web de la comisión europea llamada PVGIS (*Photovoltaic Geographical Information System*). Esta web ha sido nuestra referencia elegida para la obtención de información histórica de rendimiento fotovoltaico, radiación solar y datos meteorológicos como humedad o temperatura. A pesar de que se trate de una web muy completa que provee datos de buena calidad, no contiene una API por la cual pudiéramos acceder a los datos de una forma programática convencional, por lo que tuvimos que realizar *web scraping* para la obtención de estos datos.

La web PVGIS no tiene una extracción de datos básicos muy compleja, pero si se le quiere dar al usuario la posibilidad de poder seleccionar todas las opciones posibles de configuración que encontramos en la web, la complejidad de este proceso aumenta considerablemente. Para realizar este *web scraping* usamos dos tecnologías muy populares en este dominio:

- **Selenium:** Es un entorno de pruebas de interfaces de usuario para aplicaciones basadas en la web. *Selenium* provee una herramienta de grabar/reproducir para crear pruebas en base a la interacción del usuario con un entorno gráfico. Aunque esta herramienta está destinada a la creación de pruebas, se puede usar para simular interacción de un usuario con un sistema. En el proyecto es la tecnología encargada de toda la interacción con la plataforma PVGIS, desde el inicio hasta la descarga de los datos solicitados.
- **Beautiful Soup:** Es una biblioteca de *Python* para analizar documentos *HTML*. Esta biblioteca crea un árbol con todos los elementos del documento que puede ser utilizado para extraer información. En el proyecto se ha usado como apoyo a *Selenium*. Con ella, hemos extraído secciones de gráficas para poder tratarlas correctamente en nuestra plataforma.

Antes de la realización del *web scraping* y debido a que es un proceso largo, en la plataforma se dispone de un archivo de configuración de tipo *YAML* que actúa como registro caché para conocer si se ha realizado la petición previamente. Este archivo dispone de información de todas las peticiones genéricas realizadas y, si la petición solicitada existe, busca directamente sobre el directorio local donde se hubieran alojado esos datos previamente, sin tener que realizar la extracción de datos. En algunos casos particulares, se piden configuraciones muy específicas sobre las placas solares (inclinación, ángulo de orientación o azimut, tipo de montaje, entre otros parámetros). En estos casos, no es eficiente guardar la salida por la gran cantidad de combinaciones posibles, por lo que se realiza directamente *web scraping*, sin consultar en la caché.

Al comenzar el proceso, realizando *web scraping* sobre la plataforma PVGIS, es necesario gestionar los mensajes emergentes iniciales del sistema, como las *cookies*. Tras esto, es necesario insertar las coordenadas del centro para el cual se quiere realizar la consulta. Como las ubicaciones de los centros son conocidas, ya que se dispone de latitud y longitud de éstos, la interfaz de nuestra plataforma las introduce directamente sobre el mapa de PVGIS al seleccionar el centro en un desplegable. Tras la obtención de la ubicación es necesario especificar qué ventana temporal deseamos obtener: datos mensuales, datos diarios o datos horarios. Cada uno de estas pestañas tienen una configuración y etiquetas web diferentes.

A continuación exponemos las diferentes consideraciones que tiene cada sección y cuales se han introducido en el proyecto:

- **Datos mensuales:** se puede seleccionar ángulo óptimo para las placas solares, el cuál será calculado por la plataforma, o introducir el ángulo que se desee. Además, es posible especificar la temperatura media, que afecta directamente al rendimiento de las placas solares.

Figura 5.3: Web scraping con Selenium sobre PVGIS para obtener datos históricos mensuales

- **Datos promedio diarios:** En este análisis se obtiene un promedio diario de 2005 a 2016 de cada mes separado por horas, es decir, se obtiene el consumo promedio de las veinticuatro horas del día para todo el mes y todos los años. Se puede especificar el ángulo que se desee o el establecido por defecto (35 % en Inclinação y 0 % en Azimut).

Figura 5.4: Web scraping con Selenium sobre PVGIS para obtener datos históricos diarios

- **Datos horarios:** Esta sección de la plataforma PVGIS es la más completa, ya que permite seleccionar un elevado número de configuraciones. Se puede especificar si se selecciona inclinación y orientación óptimas o introducir los ángulos que se deseen. Permite seleccionar el tipo de montaje y la potencia de FV con la tecnología usada, potencia FV instalada( $kWp$ ) y pérdida del sistema(porcentaje). El fichero *JSON* obtenido nos ofrece información completa de todas las horas del año.

The screenshot shows the 'DATOS HORARIOS DE RADIACIÓN' (Hourly Radiation Data) configuration page. It includes a header with a sun icon and a question mark. The form contains several sections: 'Base de datos de radiación solar\*' with a dropdown set to 'PVGIS-SARAH'; 'Año inicial:\*' and 'Año final:\*' both set to '2005'; 'Tipo de montaje:\*' with radio buttons for 'Fijo' (selected), 'Eje vertical', 'Eje inclinado', and 'Dos ejes'; 'Inclinación [°]' set to '(0-90)' and 'Azimut [°]' set to '(-180-180)'; checkboxes for 'Optimizar inclinación' and 'Optimize slope and azimuth'; a 'Potencia FV' checkbox; 'Tecnología FV' set to 'Silicio cristalino'; 'Potencia FV pico instalada [kWp]' set to '1'; and 'Pérdidas sistema [%]' set to '14'. At the bottom, there is a checkbox for 'Componentes de la radiación'.

Figura 5.5: Web scraping con Selenium sobre PVGIS para obtener datos históricos horarios

Una vez seleccionada toda la configuración especificada, se obtiene un fichero *JSON* que se almacena en el sistema y se inserta en el registro caché si procede, para que se use posteriormente sin la petición de los datos en PVGIS.

### 5.2.3. Simulación de datos en tiempo real

En el caso del proyecto, carecemos de una fuente de datos establecida por la universidad para obtener datos en tiempo real, por lo que construiremos un conjunto de datos simulados para el desarrollo. Dado que el proyecto fue concebido para que la plataforma fuera desplegada en un entorno real, construiremos un mecanismo de ingesta de datos robusto y sencillo que facilite la adaptación a las diferentes fuentes de datos. Previamente al uso por parte del sistema de la ingesta de datos disponemos de dos fuentes de datos a simular, una de la que disponemos datos históricos que se han usado en otras funciones del sistema y otra que hay que generar desde cero. En ambas fuentes de datos se ha seguido la misma estructura de datos y consistencia a los datos históricos, por eso se ha decidido seguir usando el formato *CSV* y formatos de las diferentes columnas de los datos históricos de consumo de la Universidad Complutense de Madrid.

A continuación se amplía como se ha generado cada una de las fuentes de datos de origen para el caso de uso en tiempo real.

#### Generación de datos de consumo

Los datos de consumo para un determinado centro ya están disponibles en el proyecto y han sido proporcionados aunque sean datos históricos, como el enfoque de la monitorización en el proyecto es la contracción de toda la plataforma para su funcionamiento completo y en un paso posterior contactarlo a un sensor real, se ha usado una muestra de estos datos para poder implementar esta funcionalidad. En los datos de consumo se gestiona toda la línea para que sea procesada por *Spark Structured Streaming* pero este solo se queda con las columnas que son útiles para la monitorización:

- **Centro:** Nombre del centro usado como identificador y saber a que centro pertenece el dato.

- **Fecha:** Día y hora en el cual se capturaron los datos, usado para poder cruzar con la ocupación extraído en ese mismo momento.
- **Consumo:** Parte principal del estudio en la monitorización.
- **Coste:** Dato de coste mensual que no se usa directamente pero útil para análisis monetarios.

En el proyecto ha querido mantener siempre que sea posible la integridad de los fichero de origen y sus tiempos de extracción, pretendiendo adaptarnos a como son estas fuentes de datos y a como pueda ser una integración real en tiempo real futura.

### Generación de datos de ocupación

En el caso de los datos de ocupación no disponemos de ninguna fuente histórica ni en tiempo real a la cual acudir para obtener los datos, por ello a diferencia del consumo necesitamos simularla completamente. Se realiza este proceso para cada una de las variables del conjunto de datos a simular. Antes de generar los datos es necesario saber que columnas y con que formatos las necesitamos:

- **ID:** Número de identificación de esa extracción de datos. Numérico entero.
- **Centro:** Nombre del centro en el cual se han capturado los datos. Mismo formato que los datos de consumo para cruzar el mismo centro.
- **Fecha:** Día y hora en el cual se capturaron los datos. Mismo formato que los datos de consumo para cruzar la mismo fecha.
- **Num\_Occupation:** Cantidad de personas que se encuentran en el centro en ese momento. Numérico entero.
- **Num\_Students:** De la columna de ocupación cuantos son estudiantes. Numérico entero.
- **Num\_Staff:** De la columna de ocupación cuantos son personal de la universidad como profesores, mantenimiento, limpieza... Numérico entero.

Existen muchas técnica de generación de datos aleatorios; en este módulo hemos decidido utilizar una distribución normal como función generadora de datos de ocupación. En el proyecto se podrían usar dos momentos en lo que generar la simulación, bien generar los datos en el momento de lectura por parte de *Spark Structured Streaming* con un código que realice la ecuaciones oportunas y genere el dato aleatorio determinado o realizar la generación de un archivo *CSV* completo con gran cantidad de datos simulados y usar el mismo sistema de inserción que los datos de consumo. En el proyecto hemos usado la segunda opción buscando que toda la extracción de datos simulada sea igual, tanto de consumo como de ocupación y para poder disponer de los datos en un fichero por si fuera necesaria cualquier estudio o modificación.

Una distribución normales (o gaussiana o Gauss o Laplace-Gauss ) de distribución muy conocida y estudiada, es un tipo de distribución de probabilidad continua para un valor real aleatorio . La forma general de su función de densidad de probabilidad es:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-(x-\mu)^2/2\sigma^2} \quad (5.1)$$

El parámetro  $\mu$  es la media o expectativa de la distribución (también su mediana y moda), mientras que el parámetro  $\sigma$  es su desviación estándar. Se ha usado la distribución normal solo para generar la columna de Num\_Occupation, con una media y desviación estándar en 2. El valor de la  $x$  es introducido con una función de números aleatorios, que devuelve un valor entre 0 y 1. Es necesario controlar también que el valor sea absoluto ya que la distribución normal no asegura que el valor sea positivo y en el caso de la ocupación esto no es posible. Para simular la variable Num\_Students se ha restado un valor aleatorio menor a Num\_Occupation y mayor que cero. Por último, para la columna de Num\_Staff es suficiente con restar al valor de Num\_Occupation el Num\_Students, de esta forma nos aseguramos que la suma de ambas columnas no supera la ocupación del centro en ese momento. Como se ve en la figura 5.6 se muestra una gráfica con cada valor simulado de la ocupación y ordenados ordenados de menos a mayor. Se observa como sigue la distribución normal esperada, simulando menos valores en los extremos, como por ejemplo cero o nueve y una cantidad mayor cerca de la media propuesta, el dos.

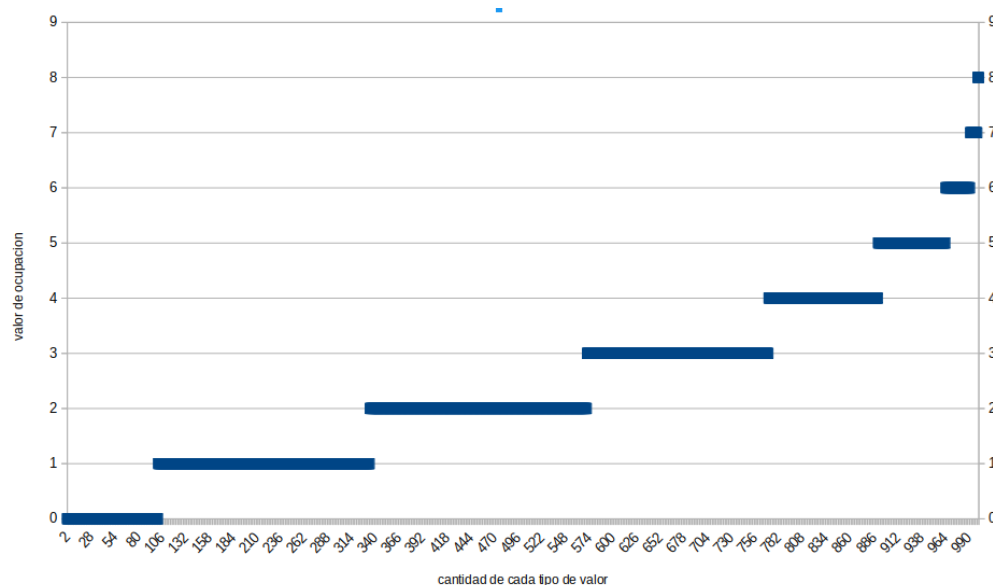


Figura 5.6: Muestra de cantidad de cada número en los datos simulados de ocupación

### 5.3. Monitorización en tiempo real con Spark Structured Streaming

El procesamiento de datos en tiempo real dista en algunos aspectos con respecto al resto de módulos del proyecto donde realizamos tratamiento de datos. Las diferencias mas destacables son las siguientes:

- A diferencia del resto de módulos, donde usamos únicamente la estructura de datos *dataframe* de la librería *pandas*, para esta funcionalidad utilizamos también *dataframe* de *Spark Structured Streaming*, como hemos comentado anteriormente. Esta estructura de datos es traducida en ambas direcciones con el resto de la aplicación. Cuando este módulo recibe una petición, convierte un *dataframe* de *pandas* a un *dataframe* de *Spark Structured Streaming* y viceversa cuando devuelve una respuesta. Éste *dataframe* puede recibir datos continuamente debido a la naturaleza del procesamiento en tiempo real, a diferencia del de *pandas*, que es discreto.
- Las respuestas son dinámicas, siempre que el *dataframe* reciba suficiente entrada. Utilizando el mismo sistema de respuestas que el resto de módulos, esto se consigue enviando múltiples respuestas en un intervalo de tiempo muy corto; cada vez que el módulo reciba datos podrá actualizar sus resultados con una nueva respuesta que será agregada a la anterior en el propio código del módulo[8, 7].

El tratamiento de datos en el contexto del procesamiento en tiempo real es similar estructuralmente al procesamiento por lotes o *batch*. En el procesamiento por lotes, se compone un conjunto de datos de entrada siguiendo una política establecida previamente. Esta política puede requerir, por ejemplo, un intervalo de tiempo entre conjunto y conjunto o, en otro caso, un número de elementos en el conjunto. En el caso del procesamiento en tiempo real, los conjuntos de datos son actualizados continuamente con datos nuevos y, al mismo tiempo, consumidos. La estructura que permite esta entrada y salida de datos continua recibe el nombre de *stream* de datos. Esto requiere de consideraciones en el código para, entre otras cosas, asegurarnos de que cuando consumimos datos del *stream*, éste no está vacío[20].

Realizando estas consideraciones, el resto del tratamiento mantiene la misma estructura de un proceso ETL(*Extract - Transform - Load*) por lotes. Un proceso ETL divide el procesamiento de datos en tres fases: ingesta de datos, transformación y carga de datos en un almacén de datos para posterior análisis. El principal objetivo de esta división es desacoplar el código de cada una de las fases, de tal forma que sea sencillo componer diferentes flujos de ejecución con combinaciones entre las posibles funcionalidades de cada fase.

### 5.3.1. Fases de la ETL

Las fases de ingesta y transformación de datos se ajustan adecuadamente a nuestro caso de uso de procesamiento en tiempo real, pero la carga de datos no, ya que no utilizamos ninguna clase de almacenamiento persistente de los resultados, por lo que sustituimos esta etapa por la explotación de datos. De esta forma, tras realizar la transformación prepararemos los resultados que enviaremos al cliente en la fase de explotación de datos a partir de un dashboard en tiempo.

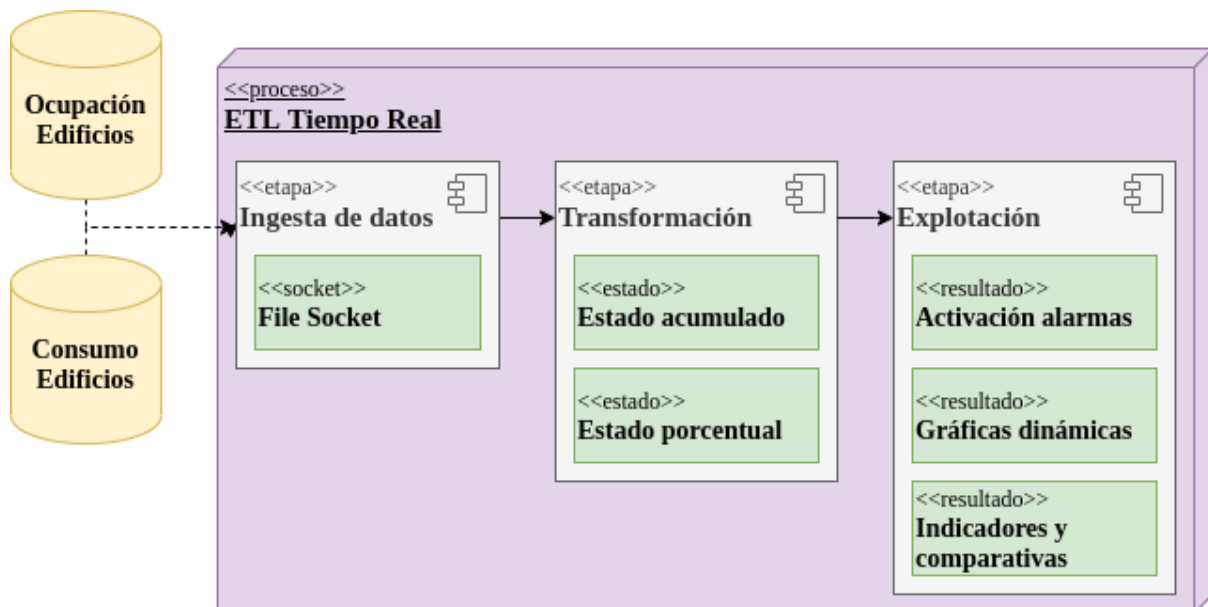


Figura 5.7: Diagrama de flujo de procesamiento de datos de monitorización en tiempo real

En la figura 5.7 podemos ver el flujo de ejecución y de datos entre las diferentes etapas de la ETL. A continuación explicaremos qué acciones tienen lugar en cada una de ellas.

### Ingesta de datos

La ingesta de datos se realiza consumiendo datos a través de dos fuentes. La primera de ellas, nos proporciona el conjunto de datos de ocupación de los edificios generado por simulación. La segunda de ellas, nos proporciona el conjunto de datos de consumo por centros proporcionado por la universidad. Estos datos los recibimos y almacenamos primero en un *dataframe* de *Spark Structured Streaming* y después se pasa a la fase de transformación de datos. Cada vez que estas fuentes de datos tengan datos nuevos, serán leídas una a una y este *dataframe* será actualizado con los últimos datos. Cuando las siguientes fases de la ETL consuman datos de esta estructura de datos, desaparecerán, ya que no se persiste el estado de la información en esta fase. Por lo tanto, en cada etapa se utiliza un *dataframe* para desacoplar las responsabilidades de cada una de ellas, creando esta estructura de datos en esta fase.

La lectura de los datos de las diferentes fuentes al *dataframe* de *Spark Structured Streaming* se realiza a partir de un socket que se comunica por el protocolo de internet TCP/IP. La ingesta de datos, al usar el protocolo TCP (*Transmission Control Protocol*), tiene una orientación a conexión. De esta forma, se evitan los errores, se controla la omisión de datos y el orden en el que la información se ha transmitido. Para lograr esta comunicación, se asigna un puerto para realizar esta transmisión y se envían datos cada cierto número de segundos por el socket para simular el comportamiento en tiempo real de un sensor. Se ha dispuesto un script de *python* que usa librerías de comunicación para actuar de servidor, este código inicia la tubería de comunicación y espera a que un cliente se conecte, en el proyecto el cliente es el propio *Spark Structured Streaming*. Una vez se ha aceptado la comunicación entre cliente y servidor, el servidor envía cada cierto tiempo una línea de los datos desde los ficheros origen simulando el comportamiento en tiempo real, por defecto este valor temporal es una hora siguiendo los tiempos de transmisión



de los datos históricos de consumo. Cada dato es leído por el cliente y en este punto ya estamos en la segunda fase de transformación de datos.

## Transformación de datos

La transformación de datos tiene un factor relevante en la monitorización del consumo, ya que se generan cambios en los datos de origen que dan el valor y estructura necesaria para que se muestre la información como es necesario. Esta fase se ve condicionada por lo que necesita la fase de explotación, ya que en esa fase los datos deben recibirse completamente preparados para pasar por las funcionalidades pertinentes, sin preprocesamiento o transformaciones. En esta etapa se realizan principalmente dos transformaciones, enviados a la siguiente etapa junto con los datos originales provenientes de la ingesta de datos. Para que el sistema tenga una coherencia arquitectónica se ha tomado la decisión de introducir este flujo también por un *dataframe* de *Spark Structured Streaming*.

Las dos transformaciones que se realizan en esta etapa son principalmente diseñadas para gestionar estados por una serie de métricas determinadas, realizando todos los cálculos necesarios sobre los datos y almacenando el estado de cada resultado para poder realizar comparaciones y visualizaciones en la explotación de datos. Las transformaciones podrían ser realizadas en la vista con algún lenguaje de scripting como *JavaScript* debido a la poca carga matemática que tienen de forma independiente, pero se perderían las ventajas de procesamiento distribuido de *Spark Structured Streaming* y con esto velocidad de procesamiento. En esta fase es necesario que los resultados obtenidos se almacenen un día completo, este estado es necesario para las comparativas con el día anterior que se hacen en la fase de explotación.

Los dos tipos de transformaciones que se usan en el proyecto son las siguientes:

- **Valor total o acumulado:** En esta forma de tratar la información se realizan operaciones simples sobre los datos de origen, organizando la información necesaria en grupos que son específicos por métrica. La métrica más sencilla es la suma de todos los elementos que hayan sido tratados. La complejidad de estas operaciones viene dada por la gestión de estos valores acumulados para un elevando número de rangos temporales, realizando cada operación para cada uno de estos rango sin perder información.

Estas operaciones se almacenan en nuevas columnas del dataframe para cada uno de los datos y en el caso de que sea un dato que dependa de todos los anteriores se expresará el valor total hasta ese momento. Insertar de esta forma los datos transformados resulta muy útil para aquellos datasets que no tienen muchas columnas como el del proyecto, ya que facilita mucho su uso en la fase de explotación de datos.

- **Valor concreto:** En el caso de necesitar un valor mas concreto que un valor total acumulado o sumatorio, es necesario disponer de una columna que establezca el valor de los datos para una métrica dentro de un rango. Por tanto se añaden columnas que realizan operaciones discretas entre columnas para obtener un valor que cruce sus métricas, por ejemplo, el consumo o gasto monetario por persona en un momento temporal.

Aquellas operaciones que requieran de un valor o valores introducidos por parte del usuario para poder realizar el calculo si se realizan en la fase de extracción de datos, esto es debido a la propia falta de datos que debe insertar el técnico en un formulario que se le ofrecerá antes de acceder a la funcionalidad y además, si se ofreciera estos valores directamente alguna de las fases de *Spark Structured Streaming* se rompería la encapsulación de capas creada en el proyecto.

## Explotación de datos

La explotación viene definida por las estructuras y formatos de los datos que se han obtenido de la fase anterior de la ETL. Esta etapa no debería realizar transformaciones si no es estrictamente necesario, ya que su objetivo es centrarse en la visualización de datos que se le va a mostrar a los usuarios.

En esta etapa diferenciamos tres componentes que muestran resultados de formas diferentes y tienen una función establecida: activación de alarmas, gráficas dinámicas e indicadores discretos. Los tres elementos tienen acceso a todas las fuentes de la etapa de transformación, aunque no todos consuman de toda la información. Este aspecto del diseño fue impulsado por el ánimo a agregarle flexibilidad al sistema a futuras ampliaciones del sistema para incorporar más elementos analíticos en el dashboard final.

Cada elemento dispone de un mecanismo de visualización explotando al máximo el potencial analítico que se puede obtener de este componente. Estos mecanismos, dinámicos, son *HTML*, *JQuery* y *JavaScript*, los cuales son utilizados en la construcción de los elementos discretos y las alarmas. Estas tecnologías son usadas principalmente para mantener los elementos actualizados en todo momento. La gestión en esta fase del control de la actualización del dashboard de forma automática para refrescar los datos mas recientes y el control de los valores máximos en los indicadores que hacen saltar las alarmas es realizado por códigos específicos hechos en *JavaScript*, estos códigos son introducidos en la creación inicial del dashboard junto al resto de componentes. Además, realizan su propia construcción, para que tenga un sentido visual dentro de la plataforma. En el caso de las gráficas dinámicas, se usa *pyplot* para mostrar los datos en tiempo real que llegan desde la etapa de transformación. Un dashboard con dos gráficas temporales, de barras para aquellos datos discretos y de líneas para los valores continuos, tres indicadores numéricos con debajo su tendencias en forma de porcentaje para aquellos valores sumatorios y acumulados, por último, un indicador visual con varios valores porcentuales para realizar el control de las alarmas.

## 5.4. Optimización presupuestaria multiobjetivo

Como ya comentamos anteriormente, para la resolución de este problema de optimización multiobjetivo utilizamos Evolución Diferencial.

### 5.4.1. Detalles de implementación

Para la codificación del algoritmo seleccionado hemos elegido la implementación *NS-GAII*, de la librería *jMetalPy* [19] en *Python*.

Nuestra solución inicializa los parámetros del algoritmo de la siguiente forma:

- **Tamaño de población:** 100 elementos.
- **Tamaño de los hijos de la población:** 100 elementos.
- **Operador de mutación:** *PolynomialMutation* con probabilidad 0.5 e índice de distribución de 20.
- **Operador de selección:** *SBXCrossover* con probabilidad 1.0 e índice de distribución de 20.
- **Criterio de finalización:** *StoppingByEvaluations* con un número máximo de evaluaciones de 25000.

El código de la implementación de la ecuación 4.3 se puede ver en el apéndice A.1.

## 5.5. Integración con Django

La plataforma web del sistema es esencial en nuestro proyecto, ya que nos permite mostrarle con facilidad al usuario los resultados de nuestros análisis y visualizaciones, pero lo más importante es aquello que está detrás, el código fuente *Python* que realiza toda la computación necesaria para obtener estos resultados. Además, dado que el foco del proyecto se encuentra sobre estas funcionalidades y no sobre el desarrollo web necesario para presentarlas, hemos considerado que un *framework* web que pudiera facilitarnos la integración de nuestro código *Python* con la plataforma web, nos ayudaría a centrarnos en el objetivo del proyecto.

Considerando estos aspectos, decidimos buscar un *framework* web que se integrase fácilmente con código en *Python* y que aportase lo necesario para que no nos tuviéramos que preocupar demasiado por las funcionalidades *Javascript* necesarias para ofrecer nuestras utilidades al usuario. Así, elegimos *Django*, ya que es el *framework* web que funciona sobre *Python* con mayor cantidad de documentación disponible. Además, realizamos una prueba de concepto utilizándolo y consideramos que es fácil de utilizar y que su curva de aprendizaje nos sería suave y agradable.

### 5.5.1. Diseñando para Django

En el diseño de la aplicación tomamos las características de *Django* como pilares sobre los cuales apoyar nuestras decisiones para, de esta forma, construir coherentemente componentes que encajen correctamente con las herramientas proporcionadas por el *framework*. Éstas características son las siguientes:

- *Django* usa una arquitectura tipo «nada compartido» lo cual significa que puede agregar hardware en cualquier nivel, servidores de base de datos, de caché o web/aplicaciones. Utilizando esta metodología de agregación de funcionalidades, incluimos la ingesta de datos correspondiente a la generación de informes así como su procesamiento, gestión y explotación.

- Además, se distribuye con un simple pero poderoso *framework* de caché. Esto último es muy relevante en nuestro sistema ya que algunas de las peticiones que puede llegar a atender la aplicación pueden tomar bastante tiempo, algo que se solventa parcialmente de esta forma. La caché permite guardar vistas ya proporcionadas previamente así como los datos almacenados en una base de datos. Además, *Django* distingue entre peticiones dinámicas y peticiones estáticas para responder a las últimas de una manera eficiente y sencilla, sin necesidad de cargar ninguna lógica.
- El *framework* separa limpiamente componentes como la capa de base de datos y la capa de la aplicación. Esto es logrado utilizando el patrón de modelado software *modelo-vista-controlador*, el cual separa las funcionalidades de interfaz del usuario y entrada de datos con el estado del modelo de datos, su lógica y su persistencia, a través de un componente software denominado controlador. Nos hemos basados en este patrón y la estructura proporcionada por *Django* para separar correspondientemente la generación de vistas de todos los módulos que realizan la computación necesaria por detrás para obtener los datos utilizados en los informes de las vistas. Esta modularidad le aporta mantenibilidad a la aplicación ya que reduce las dependencias entre componentes; cualquier extensión o cambio futuro requerirá una cantidad muy baja de cambios en el código del sistema.

Adaptándonos a la arquitectura de *Django*, desarrollamos la interfaz de usuario, el controlador y la lógica de negocio, como explicamos anteriormente. Además, también desarrollamos un módulo que gestiona la autenticación de los usuarios.

### Gestión de usuarios

La gestión de usuarios es controlada por un sistema de autenticación. Al usuario le ofrece poder iniciar y cerrar sesión, además de solicitar acceso. Para ajustar esta utilidad a los requisitos funcionales establecidos para el sistema, también hemos incluido un conjunto de grupos que permiten distinguir qué accesos pueden realizar cada uno de los usuarios registrados en el sistema.

Los accesos permitidos para cada usuario discriminan los contenidos que pueden visualizar o los análisis que pueden ejecutar. La composición de estos grupos, como la alta de usuarios, es responsabilidad de uno o varios administradores del sistema en cuestión. Estos usuarios tienen la posibilidad de acceder a la interfaz de administración del sitio web al iniciar sesión.

Este módulo le otorga al sistema integridad, ya que permite realizar distinción entre usuarios para asegurar que nadie accederá a datos protegidos.

## 5.6. Despliegue con contenedores Docker

El despliegue del sistema requiere de una gestión de recursos muy concreta utilizando funcionalidades del sistema anfitrión y de *Docker*. Un despliegue rápido y sencillo es esencial, tanto durante el desarrollo del proyecto y su mantenimiento para probar ágilmente cambios en el código como para conseguir que el sistema sea eficiente. Además, poder anular un despliegue realizado para volver a desplegar es igual de necesario.

### 5.6.1. Integración entre servicios

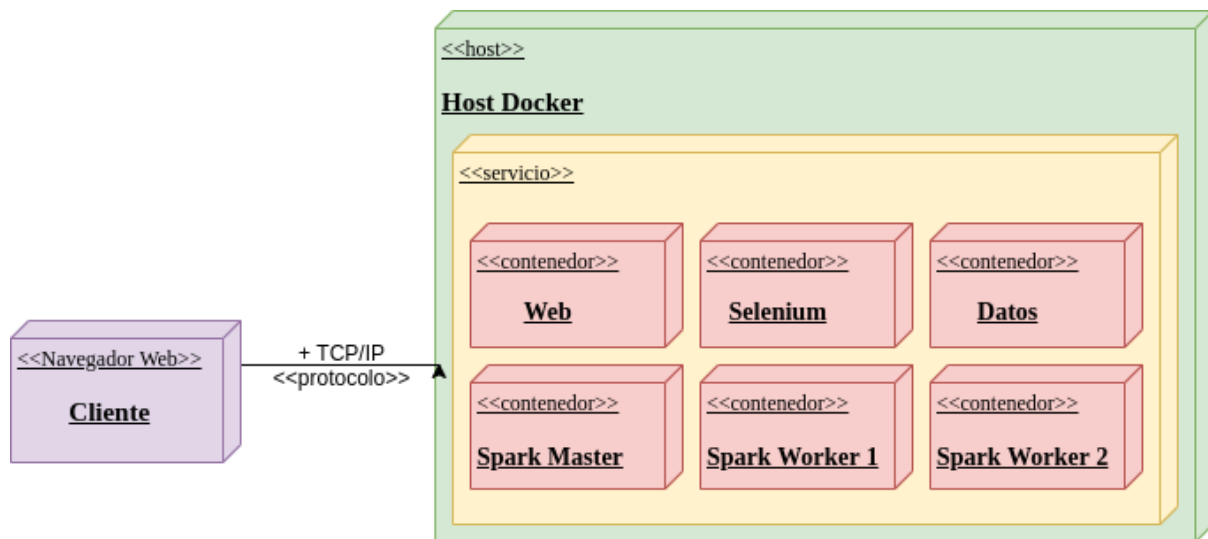


Figura 5.8: Diagrama de despliegue orquestado con Docker

La configuración del proyecto en *Docker* monta los servicios que se pueden ver en la figura 5.8, dentro del servicio externo que le ofrece *Docker* al cliente. Cada uno de los contenedores desacopla las responsabilidades de un componente software del resto. De esta forma, si fallase alguno de los contenedores, éste podría ser reiniciado o tratado sin afectar al resto del sistema. Los contenedores mostrados encapsulan las funcionalidades comentadas previamente.

### 5.6.2. Gestión del despliegue

Para lograr una gestión eficaz del despliegue del sistema hemos desarrollado dos scripts, *up.sh* y *down.sh*. Estos scripts utilizan comandos *bash* y la utilidad *docker-compose* para construir y montar los contenedores mostrados previamente, así como desmontar. La construcción y puesta en marcha de los servicios consisten en seguir las instrucciones de los ficheros *Dockerfile* que hayan sido indicados en el fichero *docker-compose.yml*. El apagado de los contenedores se basa en conocer qué contenedores de los indicados en el fichero *docker-compose.yml* están montados y desmontarlos, liberando los recursos asociados del sistema, aunque la imagen del contenedor permanece construida y guardada en la caché del sistema anfitrión.

Gracias al uso de *Docker* en el despliegue del proyecto obtenemos la portabilidad deseada en el proyecto, pudiendo realizar despliegues rápidos, sencillos e independientes entre ellos.



# Capítulo 6

## Experimentación

Los usuarios de plataformas de gestión energética, para los cuales desarrollamos este proyecto, tienen varias motivaciones para utilizar en sus proyectos estas plataformas. A la hora de realizar el mantenimiento de un edificio o valorar un presupuesto energético para tomar decisiones acertadas, acuden a software como el que desarrollamos para documentarse y resolver la incertidumbre de sus hipótesis. Consideramos que existe una motivación común en todas las necesidades y demandas de estos usuarios: valorar el ahorro y monitorizar el consumo.

En este capítulo recorreremos la aplicación valorando de qué formas ésta nos permite visualizar, medir y contrastar el ahorro, determinando de qué manera las funcionalidades del sistema son pertinentes al ahorro. Además se mostrará como el proyecto es capaz ayudar a los usuarios a monitorizar y conocer los diferentes consumos producidos.

### 6.1. Plataforma web solida

Antes de mostrar cada una de las secciones de la plataforma destinada al análisis es importante conocer sobre qué base se fundamenta el análisis. La plataforma web es igual de relevante que el resto de componentes, ya que va a ser lo primero con lo que se va a encontrar el usuario y sobre la cual va a operar; la comodidad que sienta el usuario al interactuar con la web afecta directamente a su posterior análisis. En el proyecto se busca en todo momento que la experiencia de usuario sea la más eficiente, cómoda y óptima posible.

La aplicación web como se ve en la figura 6.1 se compone de colores con tonos oscuros suaves, con un menú lateral que permite acceder a las diferentes secciones de la plataforma. En la cabecera, se encuentra la posibilidad de inicio o cierre de sesión. Esta disposición busca ser lo más sencilla posible para el usuario, ya que es popular en los sitios web actualmente. En el sistema se busca en todo momento evitar caer en el error que caen las plataformas EMS o de ámbito industrial, siendo poco intuitivas, con muchas secciones y con una curva de aprendizaje alta.

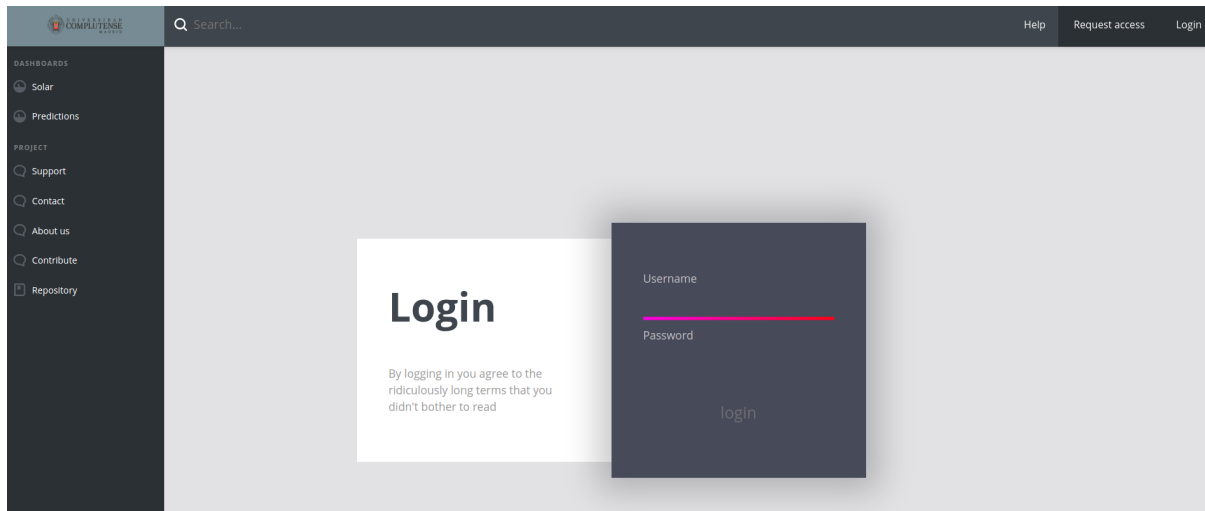


Figura 6.1: Captura de pantalla de la sección de inicio de sesión

Observando las diferentes secciones a las que podemos acceder con cualquier usuario podemos ver que en el proyecto se han dispuesto de todas las acciones que podrían necesitar los usuarios. Estas vistas son las siguientes:

- **Login and request access:** Vista como la que se ofrece en la figura 6.1 que ofrecen un formulario cada una o bien para pedir acceso o para entrar como usuario registrado si lo estas.
- **Dashboards:** Acceso a los formularios de las diferentes funcionalidades analíticas de la plataforma. A estas secciones podrás acceso si tienes permiso con tu perfil.
- **Support and Help:** Secciones donde se ofrece información valiosa para interactuar con la plataforma y preguntas típicas que puede tener un usuario.
- **About us and Contact:** En el caso que necesites contactar con los administradores o desarrolladores, existe un formulario para este propósito y además, una sección donde se ofrece información de quien ha desarrollado y respaldado el proyecto.
- **Contribute and repository:** Vista que ofrece información sobre cómo contribuir, además de un enlace directo al repositorio que contiene el código para que se pueda usar libremente, ya que un aspecto muy importante del proyecto es desarrollar software libre.

## 6.2. Análisis visual de consumo usando energía fotovoltaica

El proyecto ofrece un conjunto de informes en forma de indicadores y gráficas agregadas complejas con las que se busca ayudar a los diferentes perfiles interesados, para que, así, puedan realizar análisis para conocer los diferentes aspectos del consumo energético de un centro. Se muestra este consumo energético con el objetivo de potenciar el uso de la energía fotovoltaica. De esta forma, la plataforma acompaña al usuario a conocer, utilizando diferente información de consumo, qué sucedería con diferentes configuraciones



de energía solar.

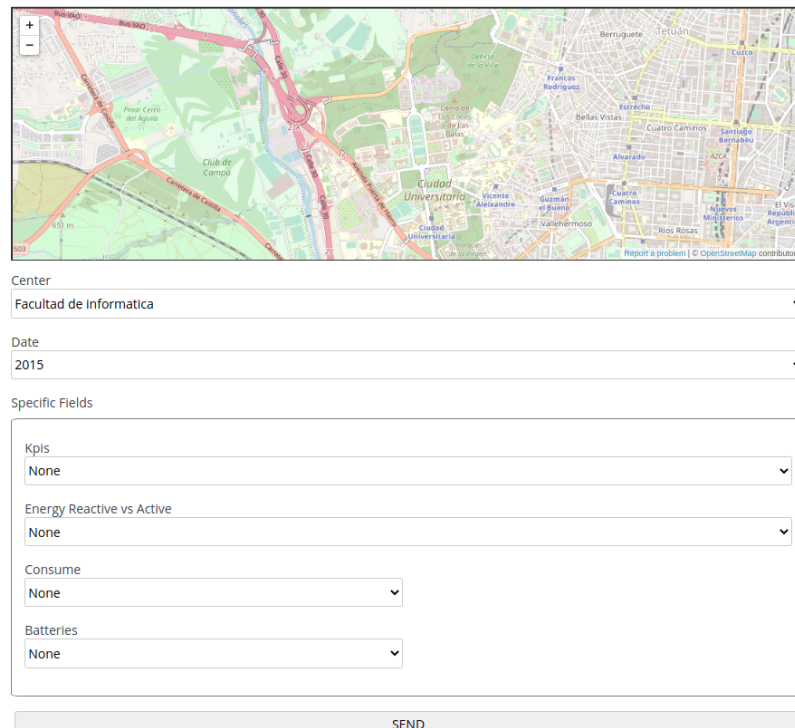
En el caso pertinente a esta sección, se va a realizar el análisis de un determinado centro utilizando toda la información de la que se podría disponer. Queremos obtener, para ese centro, información e indicativos estadísticos, comparando diferentes tipos de energías como activa y reactiva y, por último, cómo cambiaría todo esto con el uso de baterías para acumular energía.

Esta sección está destinada a los perfiles específicos de responsable de suministros, el personal de mantenimiento, estudiantes y personal docente. Además del administrador del sistema que tiene acceso a toda la plataforma. En las siguientes secciones se explican diferentes casos de estudio donde se pueden ver las funcionalidades de esta sección y qué perfil le interesa más cada una de ellos.

### 6.2.1. Formulario con configuración específica

Una parte importante de la plataforma web es el análisis de informes sobre consumo y energía solar para un centro y periodo temporal. Se dispone de un formulario específico donde el perfil puede seleccionar los parámetros que quiera ver en el informe, siempre que el perfil tenga acceso a estas secciones. En la figura 6.2 se muestra la pantalla principal de selección de parámetros disponibles para mostrar en el informe.

Solar inform request



Center  
Facultad de informatica

Date  
2015

Specific Fields

Kplis  
None

Energy Reactive vs Active  
None

Consume  
None

Batteries  
None

SEND

Figura 6.2: Formulario de generación de informes con diferentes parámetros seleccionables

A continuación, se explicará cada una de las secciones del proyecto que se pueden analizar y se dispondrá de un caso de estudio de cada apartado para una mayor comprensión de qué se pretende obtener en cada caso. El formulario dispone de más secciones ocultas

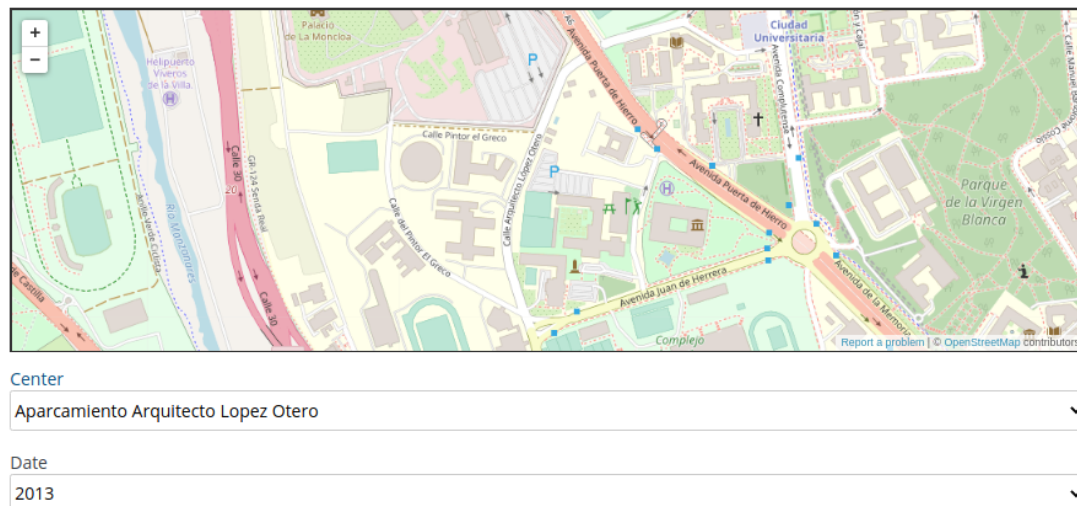
desplegables, las cuales se mostrarán con más detalle en la parte del informe a la que vayan destinados, si fuera necesario para su aclaración.

Es importante también destacar que a lo largo del informe, una vez generado, se muestra información textual interesante a nivel analítico y explicativo de cada sección. Aunque en general sirve como soporte para el entendimiento de las gráficas e indicadores, también generan conocimiento. A lo largo del informe, también es necesario, y se informa al técnico, que en la realidad nunca se dispondrán de dichas condiciones, por ser una estudio sobre datos históricos y por la cantidad de factores externos que distan al contexto de unas condiciones óptimas.

### 6.2.2. Sección de consumo genérico

Esta sección del proyecto es básica pero relevante en el análisis, ya que se muestra la información que se determina siempre por defecto y, además, los valores mínimos. Como se puede observar en el formulario de la figura 6.3, se dispone de dos campos. Estos campos especifican el centro a estudiar y el año para el cual se quiere focalizar el análisis. Encima de estos dos componentes se muestra un mapa que tiene una importancia relevante debido a que, de los centros de los que dispone la Universidad Complutense de Madrid, hay un elevado número que no son edificios, es decir, son instalaciones como aparcamientos o parques. Si se dispone de la visualización geográfica del lugar, es más sencillo conocer de un vistazo de qué tipo de centro se trata.

#### Solar inform request



Center

Aparcamiento Arquitecto Lopez Otero

Date

2013

Figura 6.3: Formulario general que permite seleccionar el año, centro y localización a analizar

En primera instancia, tras solicitar el informe, obtenemos dos tablas y dos gráficas temporales con los datos planos para su visualización rápida. Estas tablas contienen una muestra de los datos, por lo que no se puede ver la totalidad de los datos. En el caso de este estudio se han usado los orígenes de datos en un fichero *CSV* para el centro "Parking Architect Lopez Otero" el año 2013. El proyecto no tiene como foco la muestra de datos planos completos, ya que no da valor al análisis, pero una muestra si puede ser

significativa para entender las métricas que se observan en otras secciones.

Como con las tablas, las gráficas temporales permiten visualizar el consumo en kWh con detalle de un centro o la cantidad de energía en kWh que se obtenía con el uso de energía solar. En este caso, se usan los parámetros ideales de dos placas solares. Estos parámetros se componen de los grados e inclinación de las placas solares ideales para percibir la mayor cantidad de radiación del sol. La filosofía seguida en estos datos generales es poder visualizar una muestra de los datos de consumo energético del centro para un periodo de tiempo de forma rápida y una gráfica sencilla con la que puede ver el consumo, igual con la energía solar en esas mismas coordenadas.

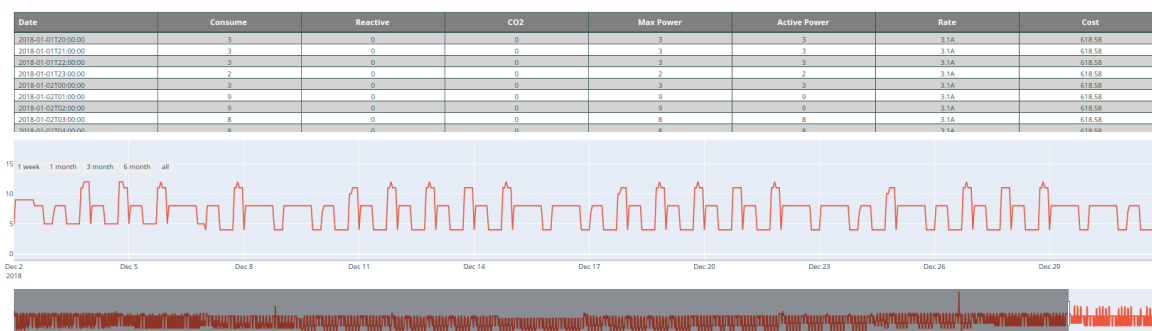


Figura 6.4: Informe con tabla de datos y su correspondiente gráfica con ventanas temporales

El rango de visualización con la gráfica es ampliamente configurable mediante el cursor para poder estudiar con el nivel de detalle que se desee la información. Se dispone de cuatro botones en la gráfica donde puedes seleccionar de forma más cómoda un periodo de tiempo determinado (semana, mes, tres meses, seis meses y todo). Por defecto, mostrará una sección final de los datos, aproximadamente un mes. Usando esta funcionalidad detectamos que parece que hay un comportamiento atípico como se visualiza en la figura 6.5, donde el consumo se eleva a un valor de 18 kWh cuando normalmente se encuentra cerca de 9 kWh. Es destacable que este valor se produce a las 3:00 de la mañana. Este comportamiento se puede visualizar de forma rápida con este tipo de gráficas temporales. De esta forma el usuario puede realizar hipótesis de lo que podría haber ocurrido.

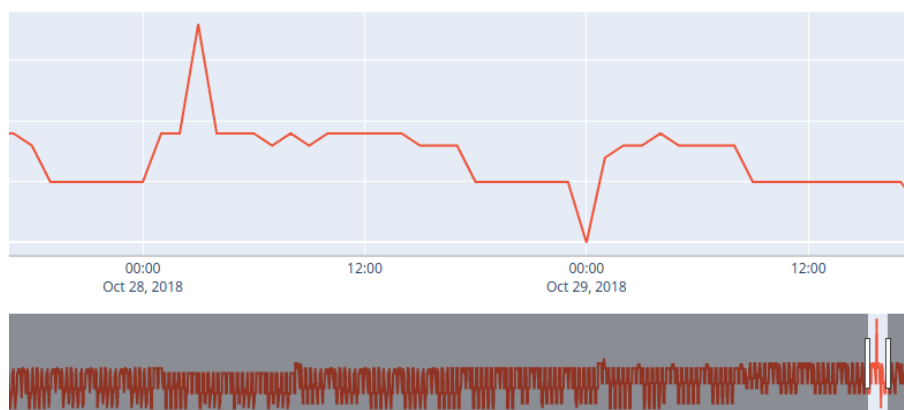


Figura 6.5: Muestra de cómo podemos estudiar datos al detalle y ayudar al usuario a sacar conclusiones

### Indicadores del centro y de energía solar

En la sección de KPIs, o de detalles sobre el conjunto de datos, se da la posibilidad de seleccionar uno de los dos orígenes de datos, los dos o ninguno. Se muestra de forma rápida información interesante de los datos vistos desde un alto nivel de abstracción, usando métricas estadísticas. Por ejemplo, la media del consumo de un centro, configuración óptima de la placa local para esas condiciones, tecnología solar usada para análisis, etc.

latitude	longitude	elevation	radiation_db	meteo_db	slope	azimuth	optimal	technology	peak_power	system_loss
40.568	40.568	2886.0	PVGIS-SARAH	ERA-Interim	29	-20	True	None	None	None

Figura 6.6: Tabla con información sobre consumo fotovoltaico de un centro

La utilización de estos datos no afecta directamente al estudio del ahorro o del consumo de un centro, pero sí ofrece información valiosa al usuario de forma que conozca metadatos y estadísticas que dan valor a un análisis energético. Como se puede visualizar en la figura 6.6, obtenemos información interesante para un técnico energético que tiene acceso a esta sección. De esta forma podría conocer qué configuración en la placa solar es la ideal para maximizar el perfil de generación.

### Energía activa y energía reactiva

En esta sección se muestra una gráfica que compara de forma simple la energía reactiva con la activa, como se muestra en la figura 6.7. Se compone de una línea que muestra la energía activa y otra línea la energía reactiva, todo esto para un centro y pantalla temporal específica.

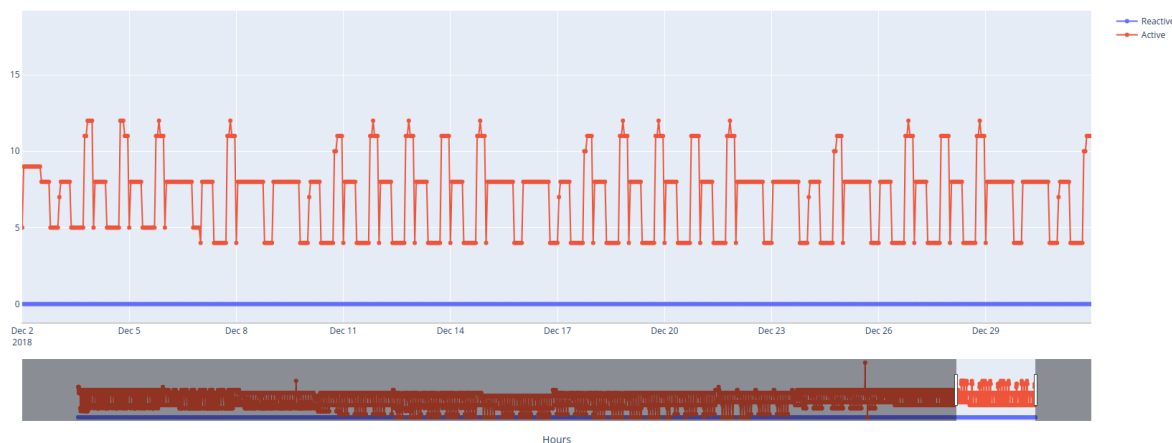


Figura 6.7: Gráfica que compara la energía activa, reactiva y máxima

Al igual que con los datos generales, se muestra por defecto una sección temporal más pequeña y reciente (normalmente 1 mes). Es importante la comparación entre las dos energías, ya que la corriente reactiva y activa son indispensables en nuestro día a día. Por esto no hay que olvidar que la energía reactiva tiene un impacto real en la factura. Un exceso de energía reactiva puede provocar todo tipo de problemas en un circuito eléctrico, que van desde la pérdida de potencia en las instalaciones hasta sobrecalentamiento en los circuitos. Es importante controlar estos indicadores para evitar estos problemas, como el exceso de consumo. Con este componente gráfico se ofrece al personal de mantenimiento la capacidad de ver si es necesario buscar una solución como la colocación de condensadores capaces de compensar la sobre-generación de energía reactiva.

### 6.2.3. Sección de gráficas complejas

En esta sección se encuentran los análisis más específicos del informe. En estos casos es necesario realizar una agregación con los datos de energía fotovoltaica para obtener la información requerida. Los datos que son usados en cada una de las gráficas e indicadores están generados respecto a los parámetros establecidos en el formulario principal de cada uno de los análisis específicos. Al igual que con los análisis generales se mostrará cada uno de los estudios ofrecidos y que ofrecen al usuario analíticamente.

#### Consumo por horas usando energía solar.

En este espacio del informe se muestra el consumo separado temporalmente en horas con dos gráficas con el mismo formato pero diferentes unidades. La primera que aparece muestra el consumo en kWh con el número de las placas solares estipulado. En la segunda gráfica se encuentra la misma configuración de placas solares pero mostrando el coste en euros. Ambas con los parámetros ideales de las placas solares. Dos gráficas interesantes analíticamente para el responsable de suministros, personal de mantenimiento, estudiante y docentes, que les permite ver con detalle y de forma directa el impacto que produce el uso de energía solar en el consumo y coste de un centro.

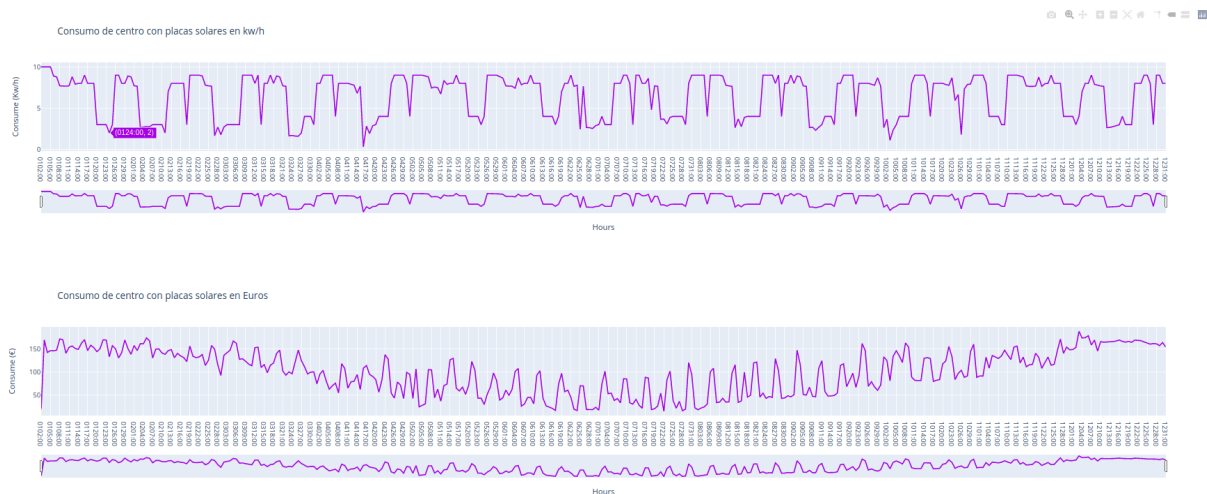


Figura 6.8: Gráficas que muestran el consumo en kWh y el coste en euros usando energía fotovoltaica a lo largo del tiempo

Estas dos gráficas, aunque simples en contexto, son claves para el estudio del ahorro de consumo y monetario usando la energía solar. En la figura 6.8 observamos que el consumo en muchas secciones temporales ha bajado a prácticamente a cero y, si realizamos una comparación directa con las gráficas de consumo sin tratar los datos, vemos cómo este consumo ha sido reducido y suavizado a lo largo de toda la gráfica. Si observamos la segunda gráfica de la figura podemos ver cómo, aunque sigue esa tendencia de picos altos y bajos en el coste, según se acercan los meses de mitad de año, se reduce el coste, esto puede ser debido a la propia actividad del centro del caso de estudio, en periodos de veranos hay menos actividad universitaria y cuando más actividad hay es en los meses de invierno. Tendencia que puede ser directa también al clima, en verano la actividad se suele hacer también fuera del centro y el invierno por posible mal tiempo se hace más actividad dentro del centro.

### Consumo por meses usando energía solar

En esta sección se muestran dos gráficas de barras que muestran una estimación del consumo por meses del centro. La primera muestra el consumo en  $kWh$  y la segunda especifica el coste en €. Cada una de ellas contiene dos métricas, una con los datos como vienen de origen y otra con uso de paneles solares, con el número de paneles solares que se estipule en el formulario y con la configuración solar ideal. El análisis final que se ofrece en estas gráficas es el mismo que en el separado por horas del punto anterior (figura 6.2.3), pero en este caso se aplica una visión más amplia que aporta de forma más sencilla un análisis de rangos temporales.

Los perfiles interesados en esta gráfica son los mismos que en el caso de la diferenciación por horas, pero en este caso se obtienen otras conclusiones. En el caso de estudio que vemos en la figura 6.9 vemos cómo se reduce el consumo y el gasto todos los meses. En diciembre, que es el mes que más gasto tiene con  $5327kW$  de consumo, se reduce a  $4910kW$ . Se aplica la configuración solar óptima. En el caso del coste en euros, en global también mejora el valor fijo contratado. En el caso de estudio de, por ejemplo, el mes de Abril, vemos que se gasta  $464.63€$  y se podría pagar con placas solares  $400.31€$ . El ahorro global vendría dado por la contratación de menos energía, debido a que con placas

solares se consumiría menos de la red contratada y directamente con esto el gasto mensual contratado sería menor. No son ahorros muy sustanciales pero en el caso de estudio solo usamos tres paneles solares, algo que debería incrementar para percibir mejor la diferencia.

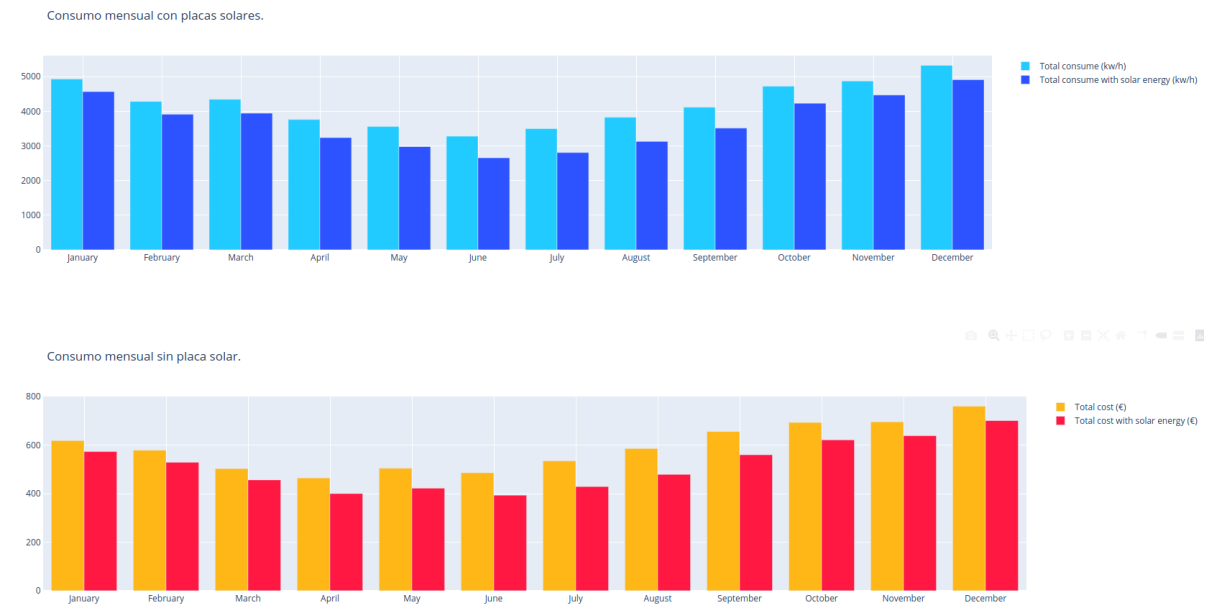


Figura 6.9: Gráfico de barras temporales donde se visualiza el consumo en kWh y euros con y sin el uso de energía solar.

### Consumo por horas usando baterías para almacenaje de energía sobrante

En este análisis nos enfocamos sobre el uso de energía solar añadiendo las baterías como un elemento más para calculo del consumo final. Se dispone de gráficas donde se puede analizar una configuración detallada de la batería y el consumo por horas de un centro en un rango de fechas estipulado. Añadir las baterías como un elemento más en el estudio del ahorro energético es esencial ya que es un elemento muy común que acompaña a las instalaciones solares para el aprovechamiento de la energía excedente en otros momentos. En la figura 6.10 se observa el consumo de forma amplia con el uso de baterías y energía solar, en este caso de estudio establecemos el uso de 10 placas solares para que la energía sobrante de energía solar sea suficiente para que se pueda almacenar en las baterías y no baje del nivel mínimo estipulado por la batería. Se ha usado una configuración de baterías como se observa en la figura 6.13, estos datos debe introducirlos el usuario con las características de la batería que quiera probar.

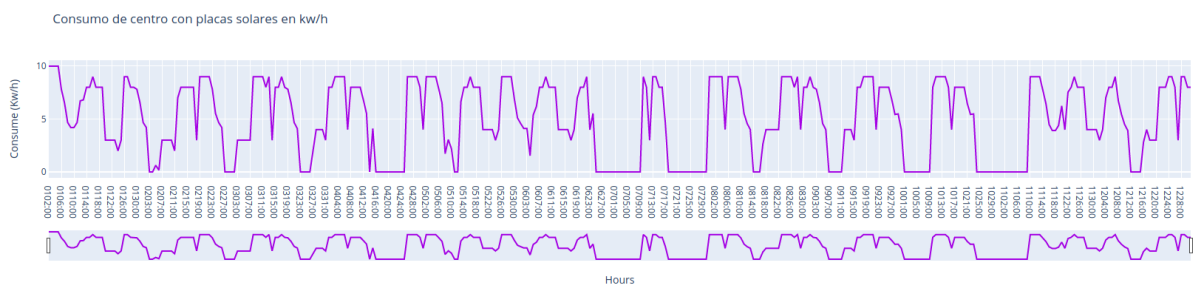


Figura 6.10: Gráfica que compara el consumo energético de un centro con el uso de baterías y energía solar.



En las figuras 6.11 y 6.12 observamos dos gráficas en detalle en un mismo periodo temporal: la gráfica superior con el uso de energía solar y apoyo con batería, la inferior solo con uso de la energía solar y ambas con el mismo numero de paneles solares (10 paneles con configuración óptima). Si comparamos ambas observamos que con el uso de baterías disponemos de periodos donde el consumo energético pasa a ser cero o menor que si no se usaran baterías como apoyo al sistema. Se observa el uso de energía excedente hasta que se consume todo lo disponible sin superar el mínimo de porcentaje permitido de descarga.

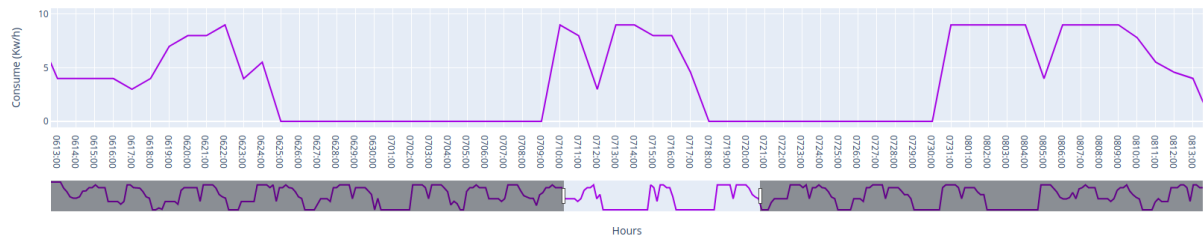


Figura 6.11: Gráfica de consumo con energía solar y baterías en un espacio temporal pequeño.

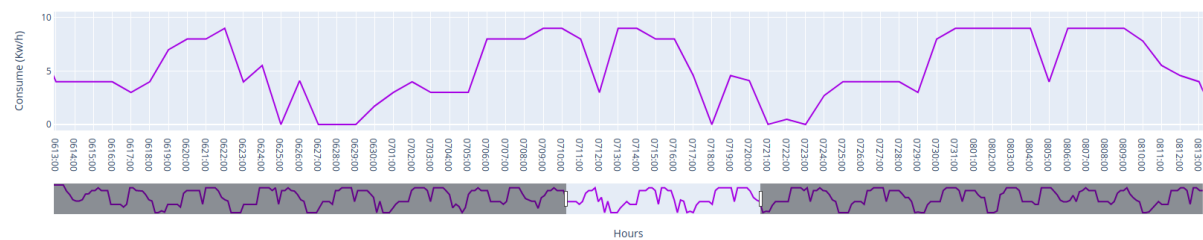


Figura 6.12: Gráfica de consumo con energía solar y sin baterías en un espacio temporal pequeño.

Se le ofrece al técnico responsable de suministros y al personal de mantenimiento, gracias a estas gráficas, el poder diferenciar rápidamente en el mismo informe la utilidad del ahorro en un inicio sin usar energía solar, después usando paneles fotovoltaicos y, por último, apoyando a la instalación de energía solar con baterías. Las baterías no suelen ser estándar respecto a estas instalaciones por este motivo se ofrece en el formulario poder especificar varios parámetros para la batería que se va a usar y que modificarán sustancialmente la información de consumo ahorrado. En la figura 6.13 tenemos el formulario previo con cada uno de los campos que puedes seleccionar para las baterías de esta sección.



## Batteries Fields

Nº Serie Batteries

2

Battery Power

100

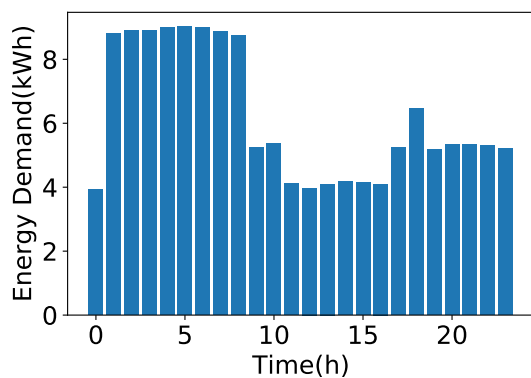
Minimal Discharging Percent

20

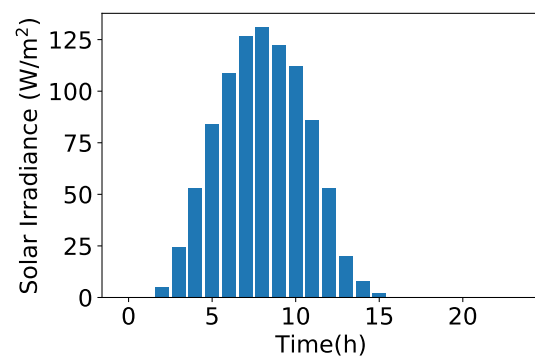
Figura 6.13: Formulario que muestra todas las opciones del análisis de consumo, energía solar y baterías.

### 6.3. Configuración óptima usando energías renovables

En esta sección mostramos ejemplos del comportamiento del algoritmo desarrollado para obtener configuraciones fotovoltaicas óptimas. La demanda energética e radiación utilizadas se corresponden con la media diaria correspondiente al año 2019 en las instalaciones ARQ. LOPEZ OTERO, véase la figura 6.14. Los datos que se han utilizado en el estudio se muestran en la tabla 6.1.



(a) Demanda de electricidad



(b) Radiación efectiva (21 % eficacia de los paneles)

Figura 6.14: Series temporales utilizando los datos de ARQ. LOPEZ OTERO del 2019

Con los datos mostrados, el coste normalizado por panel y por batería son,  $\xi_P = 1,07$  y  $\xi_B = 0,42$ . Además, el ratio de precios de venta y compra es  $\eta = 0,49$ . Por lo tanto, la solución de coste óptimo consistiría en no comprar ningún panel ni batería, como se puede ver en la figura 6.16a. La tabla 6.2 muestra la solución para diferentes criterios de optimización. La figura 6.16 muestra el conjunto de soluciones de Pareto para estos datos. Se puede ver que conectan la solución de coste óptimo y la solución de caso aislado, pero no la conectan en línea recta, sino que el número óptimo de baterías es cero hasta 50m<sup>2</sup> de paneles solares.

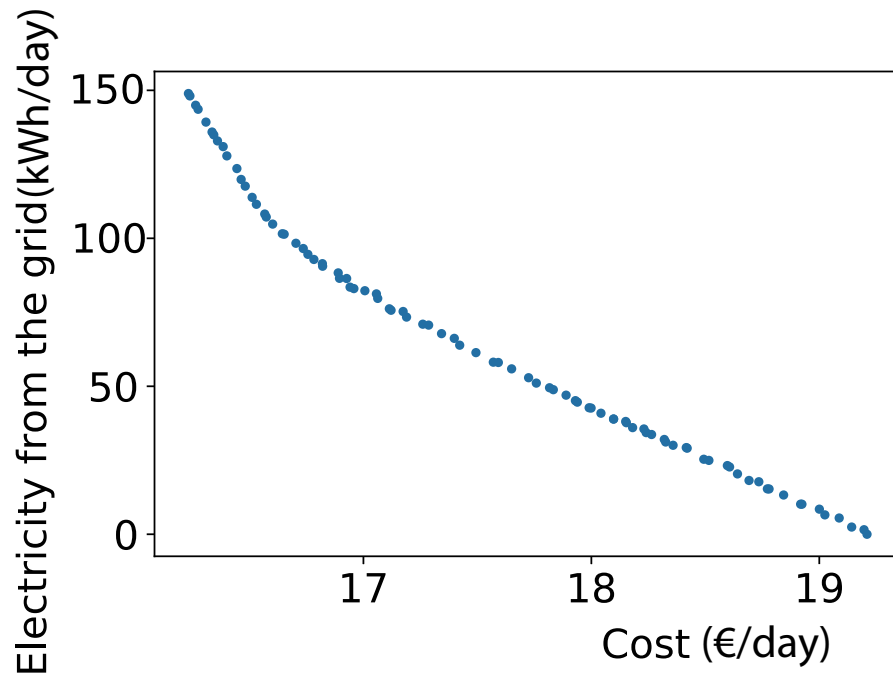
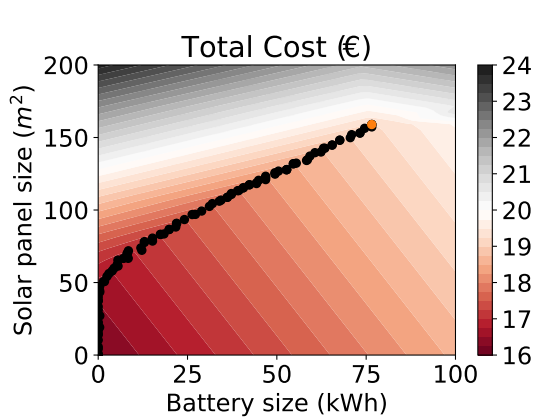
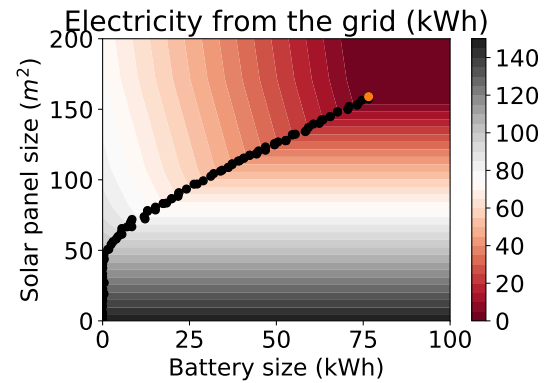


Figura 6.15: Frente de Pareto con la solución óptima de Pareto comparando el coste diario de la instalación con la electricidad consumida de la red



(a) Conjunto óptimo de Pareto en la función objetivo coste diario



(b) Conjunto óptimo de Pareto en la función objetivo cantidad de electricidad de la red eléctrica.

Figura 6.16: Funciones objetivos con el conjunto óptimo de Pareto para los datos históricos. El punto naranja es la solución de caso aislado.

Cuadro 6.1: Valores de entrada usados en el modelo energético para problema de optimización.

Variable	Simbol	Valor	Unidades
Coste de la electricidad de la red eléctrica	$C_e$	0.109[21]	€/kWh
Valor de la electricidad a la red eléctrica	$C_v$	0.054	€/kWh
Coste de los paneles solares	$C_P$	800[22]	€/m <sup>2</sup>
Coste de las baterías	$C_B$	128[23]	€/kWh
Vida útil de los paneles solares	$V_P$	20	Años
Vida útil de las baterías	$V_P$	15	Años
Perfil horario de demanda de energía eléctrica	$d_i$	Fig 6.14a	kWh
Perfil horario de generación solar	$g_i$	Fig 6.14b	kW/m <sup>2</sup>

Cuadro 6.2: Valores para diferentes soluciones en el caso de los datos de las instalaciones de ARQ. LOPEZ OTERO.

Método	Número de paneles solares (m <sup>2</sup> )	Número de baterías (kWh)	Coste (€/dia)	Electricidad de la red (kWh)
Sin baterías ni paneles solares	0	0	16.2	148.9
Sistema aislado	158.9	76.55	19.2	0
Coste óptimo	0	0	16.2	148.9

## 6.4. Monitorización del consumo comparándolo con la ocupación de un edificio

En esta sección del proyecto se mostrará la utilidad que tiene el uso de un sistema de monitorización para realizar tareas de control sobre el consumo de un determinado centro. Dentro de un sistema de estas características se dispone de paneles con indicadores y gráficas que muestran métricas de interés del técnico y sistemas de alarmas que avisan de los parámetros más críticos o importantes de controlar.

Disponemos de tres enfoques que se han introducido en el proyecto para realizar esta monitorización: el consumo, la ocupación y el cruce de ambos enriqueciendo los resultados analísticos del *dashboard*. Se explicará un caso de estudio con un conjunto de datos determinado donde se observa cada una de las funcionalidades de esta sección del proyecto al detalle.

### 6.4.1. Datos extraídos y simulados

El proyecto no dispone de un extractor directo desde un sensor en tiempo real, ni de datos de consumo ni de ocupación en tiempo real, por tanto, en esta sección del proyecto se han usado datos históricos imitando el comportamiento de extracción en tiempo real o se ha simulado totalmente si no se disponía de ellos. En el caso de los datos energéticos del consumo, como disponemos de datos históricos, se han usado directamente

los datos de los que se disponía. Para los datos de ocupación de un edificio se han simulado completamente usando una campana de Gauss con el objetivo de que sean lo más equilibrados posibles, para que sigan una tendencia lo más parecida a la real y que no se trate de número aleatorios que no permitan ver casos de estudio útiles en el proyecto.

En la información histórica encontramos una separación temporal entre cada dato de una hora. En la simulación del tiempo real se ha seguido este nivel de actualización buscando imitar lo máximo posible el comportamiento del sensor real y su posible incorporación futura. La frecuencia de extracción de los datos puede ser cambiada en el sistema en cualquier momento de forma rápida y simple en los archivos de configuración. Como vemos en las figuras 6.17 y 6.18 se observan los datos origen tanto de consumo energético como de ocupación. Se tratan de una muestra de los datos. Como los datos históricos son de 2018 el caso de estudio y la simulación de estos datos son de este mismo año. Los conjuntos de datos de muestra que se observan son contruidos de forma progresiva insertando datos en cada periodo de tiempo, como sucedería con un sensor real.

num	centro	id	fecha	consumo	reactiva	co2	potenciamax	potenciaact	tarifa	coste
0	AP ARQ. LOPEZ OTERO	78922	2018-01-01 01:00	10	0	0	10	10	3.1A	618.58
1	AP ARQ. LOPEZ OTERO	78922	2018-01-01 02:00	10	0	0	10	10	3.1A	618.58
2	AP ARQ. LOPEZ OTERO	78922	2018-01-01 03:00	10	0	0	10	10	3.1A	618.58
3	AP ARQ. LOPEZ OTERO	78922	2018-01-01 04:00	10	0	0	10	10	3.1A	618.58
4	AP ARQ. LOPEZ OTERO	78922	2018-01-01 05:00	9	0	0	9	9	3.1A	618.58
5	AP ARQ. LOPEZ OTERO	78922	2018-01-01 06:00	9	0	0	9	9	3.1A	618.58
6	AP ARQ. LOPEZ OTERO	78922	2018-01-01 07:00	8	0	0	8	8	3.1A	618.58
7	AP ARQ. LOPEZ OTERO	78922	2018-01-01 08:00	8	0	0	8	8	3.1A	618.58
8	AP ARQ. LOPEZ OTERO	78922	2018-01-01 09:00	8	0	0	8	8	3.1A	618.58
9	AP ARQ. LOPEZ OTERO	78922	2018-01-01 10:00	8	0	0	8	8	3.1A	618.58
10	AP ARQ. LOPEZ OTERO	78922	2018-01-01 11:00	9	0	0	9	9	3.1A	618.58

Figura 6.17: Ejemplo de datos de origen con información de consumo energético

id	centre	time_extract	num_occupation	num_students	num_staff
1	AP ARQ. LOPEZ OTERO	2018-01-01 01:00	4	2	2
2	AP ARQ. LOPEZ OTERO	2018-01-01 02:00	4	2	2
3	AP ARQ. LOPEZ OTERO	2018-01-01 03:00	4	2	2
4	AP ARQ. LOPEZ OTERO	2018-01-01 04:00	4	2	2
5	AP ARQ. LOPEZ OTERO	2018-01-01 05:00	3	2	1
6	AP ARQ. LOPEZ OTERO	2018-01-01 06:00	3	2	1
7	AP ARQ. LOPEZ OTERO	2018-01-01 07:00	3	2	1
8	AP ARQ. LOPEZ OTERO	2018-01-01 08:00	3	2	1
9	AP ARQ. LOPEZ OTERO	2018-01-01 09:00	3	2	1
10	AP ARQ. LOPEZ OTERO	2018-01-01 10:00	3	2	1

Figura 6.18: Ejemplo de datos de origen con información de ocupación de un edificio

Observado la figura 6.19 observamos cómo, tras el procesamiento en tiempo real con *Spark Structure Streaming*, se unen los datos en un único *dataframe* que es usado para generar las gráficas en tiempo real y ofrecer la monitorización esperada. Éste añadirá las columnas necesarias del *dataframe* de consumo energético y del *dataframe* de ocupación. Además, se añadirá al *dataframe* final los resultados de los datos procesados, información a la que se le ha aplicado toda la lógica necesaria. De esta forma la vista del *dashboard* en tiempo real solo tiene que leer los datos ya procesados y mostrarlos. El procesamiento de estos valores matemáticamente podrían ser soportados por la vista, pero es importante

aprovechar el procesamiento distribuido que tiene *Spark Structure Streaming* y así obtener los resultados con mayor velocidad que si se hubiera realizado esta lógica en la vista.

A la hora de generar el *dataframe* final se han tenido en cuenta las columnas que nos sirven como identificadores: el centro, la fecha y la hora. Estas tres columnas son cruzadas en ambos *dataframes* de origen. En el caso de estudio solo disponemos de un centro, pero el proyecto está preparado para albergar diferentes centros de estudio. Se ha agregado a las columnas de identificación de ambos *dataframes* las columnas específicas de cada uno y se han añadido seis columnas más, diferenciadas en tres parejas:

- **today\_consume y yesterday\_consume:** Columnas que contienen el valor total de consumo para cada momento del día, ya sea del día actual o del día anterior.
- **today\_occupation y yesterday\_occupation:** Columnas que contienen el valor total de ocupación para cada momento del día, ya sea del día actual o del día anterior.
- **today\_person y yesterday\_person:** Columnas que contienen el el valor de consumo en kW por persona para ese momento temporal, ya sea del día actual o del día anterior.

Las columnas añadidas a partir de transformaciones que son del día anterior aprovechan que el último estado almacenado del *dataframe* es justo la hora que se está procesando, pero del día anterior, escribiendo este dato para poder tratarlo y enviarlo a la vista del *dashboard*.

time	coste	num_occupation	num_students	num_staff	today_consume	today_occupation	today_person	yesterday_consume	yesterday_occupation	yesterday_person
010.50	4	2	2	18	4	2,50	9	4	2,50	
010.50	4	2	2	20	8	2,50	19	8	2,50	
010.50	4	2	2	30	12	2,50	29	12	2,50	
010.50	4	2	2	40	16	2,50	38	15	2,50	
010.50	3	2	1	49	19	3,00	47	18	2,50	
010.50	3	2	1	58	22	3,00	68	21	3,00	
010.50	3	2	1	66	25	2,67	68	24	3,00	
010.50	3	2	1	74	28	2,67	76	30	3,00	
010.50	3	2	1	84	31	2,67	86	32	1,50	
010.50	3	2	1	91	34	2,67	97	35	1,50	

Figura 6.19: Columnas necesarias y insertadas al dataframe base con las transformaciones hechas con Spark Structure Streaming

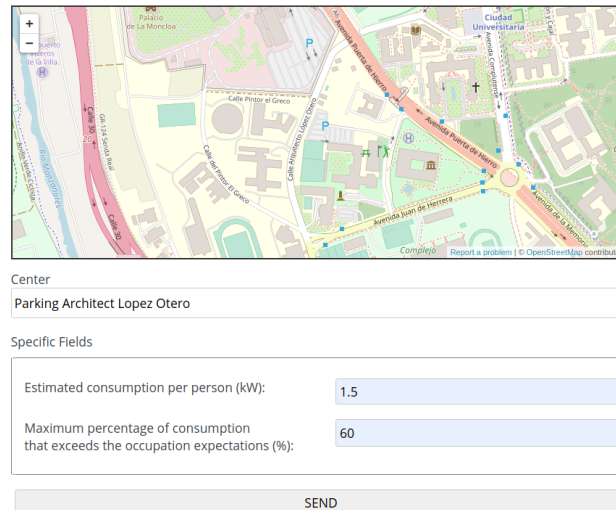
### 6.4.2. Solicitud y vista general

Para acceder al *dashboard* donde se ofrece la monitorización e información ya detallada, es necesario que se especifiquen varios valores y filtros que se aplican de forma directa en los datos mostrados en el análisis. Para obtener esos filtros se provee de un formulario con estética parecida al resto de la plataforma web. De esta forma, el usuario estará familiarizado con la interacción con este. La información a introducir por el técnico energético como se ve en la figura 6.20 es el centro de estudio, el consumo máximo en kW que se estima que genera cada persona que accede a las instalaciones y el porcentaje máximo de consumo que excede de la ocupación esperada.

El centro es un desplegable donde aparecen los centros disponibles para el análisis y que cuentan con el sistema en tiempo real disponible, además de los campos donde escribir los valores antes descritos. Estos valores deben ser numéricos, si no, se mostrará un aviso al

usuario para que corrija los datos. En el caso de estudio que nos concierne se seleccionará el centro llamado "Parking Architect Lopez Otero", con 1.5 kW de consumo por persona y 60 como valor porcentual para el máximo de consumo que excede la ocupación esperada.

### Real-time Dashboard request



Center  
Parking Architect Lopez Otero

Specific Fields

Estimated consumption per person (kW): 1.5

Maximum percentage of consumption that exceeds the occupation expectations (%): 60

SEND

Figura 6.20: Formulario de acceso al dashboard que continua la estética de la web

Una vez enviada la petición de acceso al *dashboard* con los filtros indicados se genera la vista de la figura 6.21, que compone las principales funcionalidades que se han estipulado como necesarias para este caso de uso. El *dashboard* se separa en tres secciones diferenciadas, estipuladas como tres visiones analíticas diferentes desarrolladas en los requisitos (sección 4.4). Se muestra un texto con una fecha y hora que expresan la última vez que fue actualizado el *dashboard*. De esta forma es posible conocer la ultima vez que se pudieron actualizar los datos por unos más recientes. Las tres partes del panel serán explicadas con detalle en las siguientes secciones.

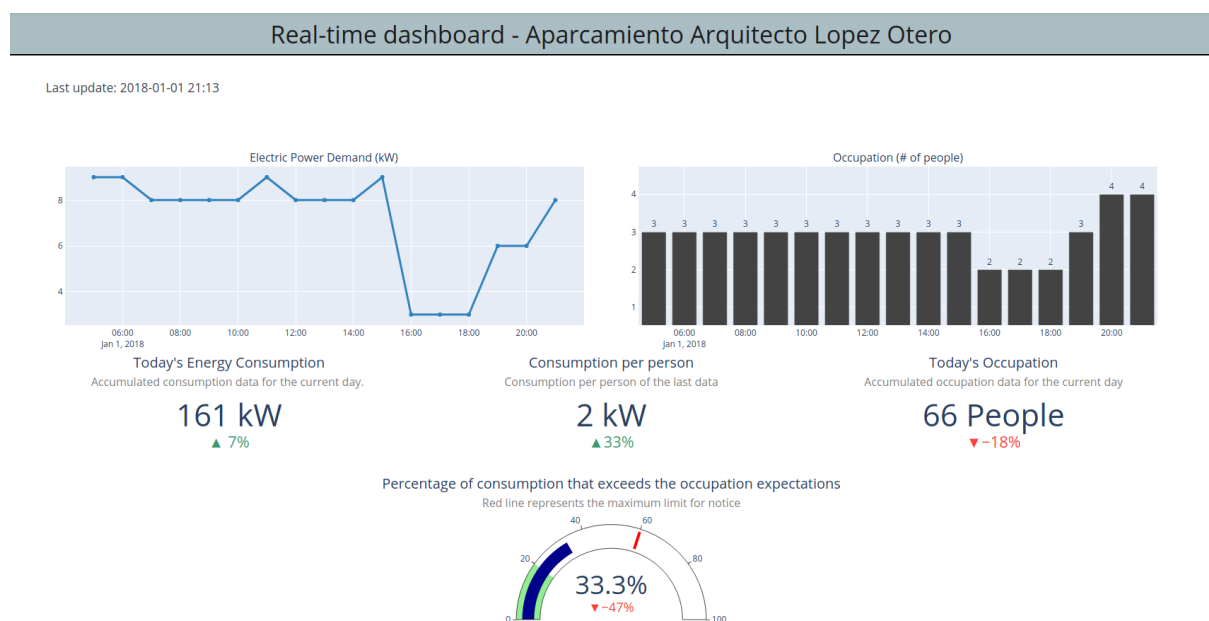


Figura 6.21: Vista general del dashboard para este caso de estudio

### 6.4.3. Gráficas en tiempo real sin transformar

Una parte muy importante de la monitorización en un sistema energético es la visualización de datos sin transformar o tratar, es decir, tal y como viene de origen. Este detalle ofrece al técnico la capacidad de conocer en todo momento el estado del sistema, como la generación de picos altos o bajos en el consumo, conocer si un sistema es inestable o si el consumo es muy alto para el momento temporal del estudio, entre muchas otras cosas que puede interpretar el profesional energético al visualizar los datos.

Para proveer de esta funcionalidad, se ofrecen dos gráficas diferentes: una gráfica temporal de líneas para el consumo energético en kW y una gráfica de barras para la ocupación en tiempo real. Ambas muestran las últimas diecisiete horas de información procesada por el sistema en tiempo real. Este número viene establecido para que ambas gráficas se visualicen con el tamaño y estética correcta, evitando sobrecargar al técnico con sobreinformación y ofrecer solo la más reciente de forma clara. Como se ve en las figuras 6.25 y 6.26, tener ambas gráficas nos permite comparar de forma directa tendencias y observar posibles picos en los datos que ocasionen una incidencia. En el caso de estudio se observa que ambas gráficas siguen una tendencia parecida, directa entre el número de personas en el centro y la cantidad de consumo que se produce. En el apartado 6.4.5, donde se explica el caso de estudio para el sistema de alarmas, se observará con más detalle el impacto en un descuadro entre estas gráficas y cómo se facilita el estudio en estos casos.

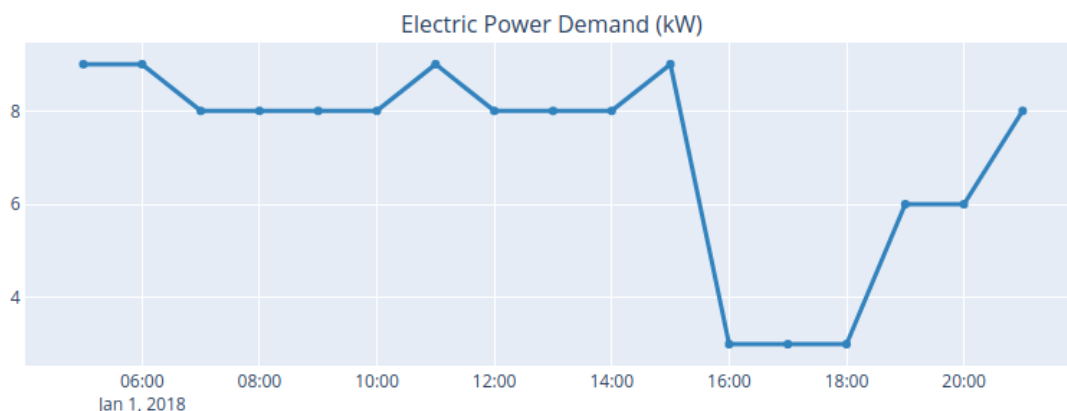


Figura 6.22: Gráfica de líneas en tiempo real que muestra el consumo en kW

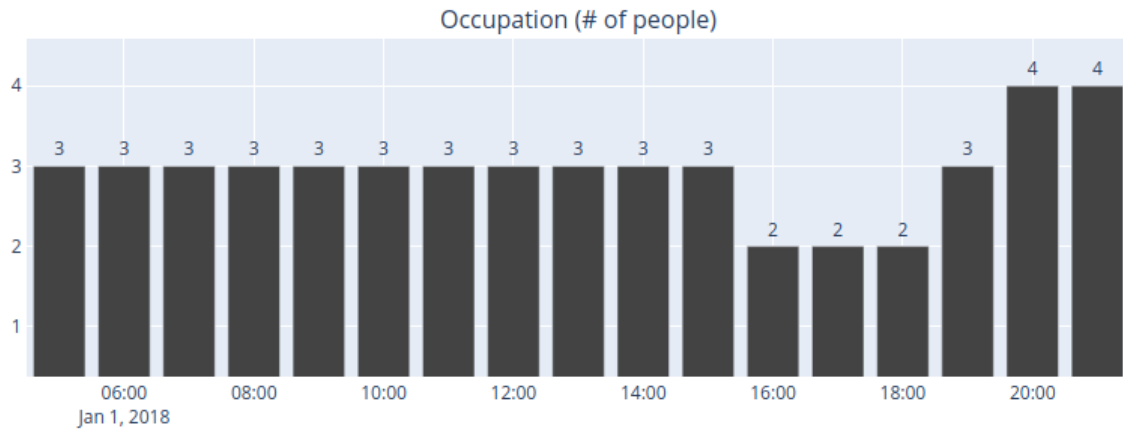


Figura 6.23: Gráfica de barras en tiempo real que muestra la ocupación del centro

#### 6.4.4. Indicadores de consumo y comparativas

El proyecto está enfocado principalmente a ofrecer la máxima capacidad analítica, además de tener gráficas en tiempo real es necesario tener valores que muestren un caso de estudio en particular. En el proyecto se provee de tres indicadores que, con la misma metodología, muestran tipos de métricas diferentes. Disponemos de tres indicadores, dos acumulativos y uno específico de valor único. En la figura 6.25 se muestran los tres indicadores específicos con los resultados para este caso de estudio, a continuación se explicarán en detalle cada uno y el por qué de sus valores:

- **Total diario de consumo energético:** Este indicador es el primero que se observa desde la izquierda y muestra la suma total de los kW consumidos en el día actual de estudio; cada día es reiniciado el valor a cero. Este indicador es muy útil para que el técnico tenga una noción de cuánto consumo se está realizando cada día y si es el esperado en cada momento.
- **Consumo por persona:** El indicador situado en medio de los tres expresa una división entre el consumo y la ocupación para conocer el valor de consumo en kW de cada persona. Con este dato, el técnico es capaz de conocer si su estimación máxima de consumo por persona estaba bien dirigida y estudiar cuál es el consumo por persona en cada momento para futuras acciones. En el caso de estudio se observa cómo se supera en 0.5 el valor que el técnico ha estimado por persona.
- **Total diario de ocupación:** Este indicador se encuentra el tercero si observamos desde la izquierda. Especifica la suma de personas que acceden al centro a lo largo del día actual. Este indicador es muy útil para hacer un control sobre la ocupación en un centro y conocer si se encuentra entre unos valores esperados o se encuentra por encima o por debajo.

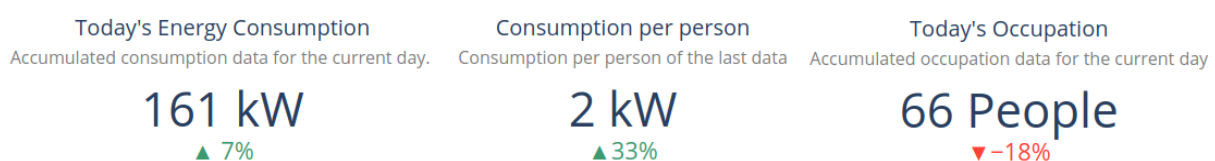


Figura 6.24: Los tres indicadores disponibles en el *dashboard*



Para la ayuda del análisis en los tres indicadores del *dashboard* se ofrece en la parte inferior de cada uno un porcentaje de tendencia que expresa cómo se encuentra ese valor actual con respecto a esta misma métrica y para este mismo momento temporal en el día de ayer. Para el caso de estudio vemos que puede surgir un problema en las tendencias debido a que el consumo total está subiendo con respecto a ayer, en cambio la ocupación total está bajando, representando un posible consumo mayor del esperado esta franja horaria.

#### 6.4.5. Alarma de consumo con respecto a la ocupación esperada

Un sistema de alarmas establecido sobre los elementos principales o críticos de un sistema genera una seguridad y control sobre un técnico energético, eso es debido a que le ofrece una forma de conocer que los componentes se encuentran entre unos límites marcados. En el caso del proyecto, al no tratarse de sistemas críticos vitales, si se superan los parámetros establecidos se avisará con una notificación llamativa en pantalla que permanecerá hasta que pase un día desde el aviso o hasta que el usuario la cierre haciendo *click* en ella, un aviso simple pero efectivo

El componente que contiene la alarma se puede observar en la figura 6.25 y se trata de un indicador como porcentaje de consumo con respecto a la ocupación esperada. El rango está establecido de 0 % a 100 % y se notifica cuando sobrepasa un valor porcentual el consumo en comparación a la ocupación que hay en un edificio. Por tanto, es necesario que el técnico especifique cuál es el límite en porcentaje sobre lo que para su criterio profesional es el máximo que debe generar preocupación y hacer saltar la alarma.

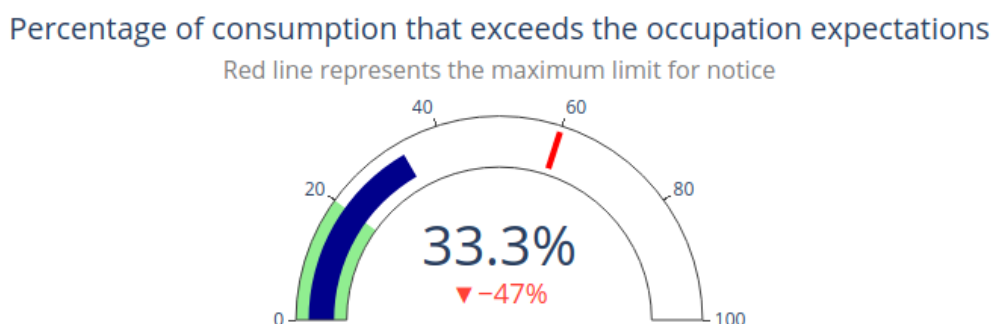


Figura 6.25: Indicador de control dentro de los parámetros normales establecidos por el técnico en el formulario

En la figura 6.25 se observa cómo se tratan parámetros normales usando el caso de estudio de toda la sección, con los datos introducidos en el formulario previo. En este indicador observamos cómo tenemos un 33.3% de consumo más del estimado para la ocupación actual, aproximadamente la mitad del límite máximo que ha estipulado el técnico. En el componente también se muestra el porcentaje respecto al valor anterior. En el caso de estudio ha bajado un 47% con respecto a la media. El dato de tendencia permite observar que, aunque estaba muy alto este valor esta tendiendo a la baja.

En el siguiente dato obtenido por el *dashboard*, se recoge un consumo en kW de 10 y podemos observar que el indicador de la ocupación continúa en 4 personas. Se produce

un incremento muy elevado del consumo para el mismo número de personas en el centro. Este detalle hace que aumente severamente el porcentaje de control y haga saltar las notificaciones oportunas. En la figura 6.26 se observa cómo supera el 60 % máximo establecido por el técnico y en la figura 6.27 la notificación que se muestra en la pantalla. Además, observamos que respecto a la media de los datos procesados de hoy ha aumentado un 3 %, algo que puede indicarnos que estamos normalmente cerca del máximo que ha establecido el técnico.

### Percentage of consumption that exceeds the occupation expectations

Red line represents the maximum limit for notice

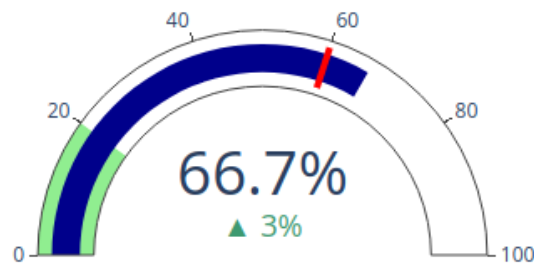


Figura 6.26: Indicador que muestra cómo se ha superado el umbral por un consumo superior al esperado

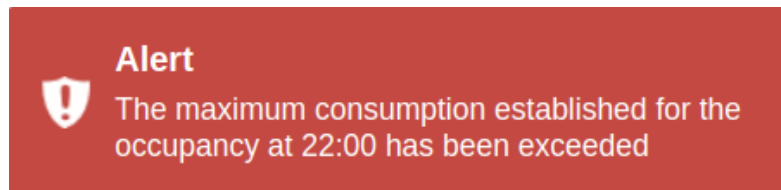


Figura 6.27: Ejemplo de alarma producida con el dato recibido a las 22:00PM

La notificación que se muestra en la figura 6.27 aparecerá en la parte superior de la pantalla y permanecerá durante un día; el usuario puede cerrarla pinchando en ella en cualquier momento.

# Capítulo 7

## Conclusiones y trabajo futuro

### 7.1. Conclusiones

A la vista de los resultados obtenidos y del desarrollo del proyecto, podemos determinar varias conclusiones. La primera es la importancia que tiene un sistema de gestión energética (EMS) en cualquier organización por el conocimiento que es capaz de generar sobre cómo está siendo usada la energía en sus instalaciones, cómo se podrían ahorrar gastos y cómo se podría contribuir a ayudar al planeta produciendo menos  $CO_2$ . En la época actual, están ganando mucha importancia las energías renovables, generando así una dificultad añadida al control y monitorización energética de un centro. Esto hace más importante el uso de un sistema como este.

Ha resultado notoria también la importancia que existe en la creación de herramientas libres en el ámbito energético, ya que existen muchas opciones que componen la mayoría o todas las funcionalidades que una organización puede necesitar pero suelen ser muy caras y con licencias privadas. Es importante disponer de una plataforma que cubra las necesidades de una organización y que permita a la comunidad mejorar el sistema, algo que ha funcionado muy bien en otros campos. El repositorio del proyecto puede ser accedido navegando a la dirección [www.github.com/albertopastormr/tfg](http://www.github.com/albertopastormr/tfg). El proyecto ha sido publicado con licencia *GNU General Public License v3.0*.

La creación de esta plataforma tiene que realizarse con una mentalidad claramente desde el punto de vista del usuario, ya que los proyectos empresariales tienden a olvidarse de controlar la experiencia de usuario y provocan que éstas herramientas industriales de esta índole sean muy difíciles de usar. El proyecto muestra cómo se puede construir un sistema que, mostrando un fuerte componente analítico, es capaz de dar al usuario final un diseño e interfaz simple, intuitivo, útil y fácil de usar.

La plataforma construida en el proyecto demuestra cómo se pueden agrupar y satisfacer las necesidades analíticas de varios perfiles, disponiendo de las funciones que necesita cada uno y utilizando un control de seguridad para que cada uno observe solo la información para la cual tiene acceso. Este componente, además de ayudar en aspectos de privacidad de datos, constituye un perfil dentro de sus intereses, evitando así saturar con información que no necesita el usuario.

Conociendo la necesidad que tiene cada perfil, el proyecto es capaz de componer análisis históricos que, unidos a información de estimaciones de uso de energía fotovoltaica, genera informes completos sobre aspectos críticos a estudiar sobre el consumo y ahorro energético de una organización. El sistema también permite conocer cuál es la configuración óptima de una instalación fotovoltaica a partir del consumo deseado y la localización, ofreciendo el mejor posible presupuesto y combinación de elementos para el sistema. Este factor es muy útil para un técnico por el apoyo que se le ofrece en la toma de decisiones, sin la disposición del proyecto, debería estudiar todas las combinaciones de los posibles componentes que puedan ajustarse al sistema. Finalmente, la plataforma permite monitorizar un edificio, mostrando si el estado energético y consumo actuales están dentro de los parámetros que han sido estipulados para una instalación energética. Además, gracias a la unión de los datos de ocupación de esa instalación, la plataforma provee de un sistema de alarmas que permite controlar las métricas más críticas o importantes.

En conclusión, el proyecto les ofrece a los responsables de las instalaciones energéticas de la Universidad Complutense de Madrid un EMS específico con el que reducir la incertidumbre en sus tomas de decisiones, monitorizando el sistema, realizando análisis energéticos de datos históricos y de datos en tiempo real. La simplicidad de la plataforma, junto con la gestión de los datos, demuestran que, dado el enunciado que planteaba resolver el proyecto, se ha obtenido un sistema efectivo.

## 7.2. Trabajo futuro

La plataforma ha sido desarrollada con la flexibilidad y la usabilidad en mente para que, a través de la publicación de la misma como software libre, cualquier institución, empresa o particular pueda utilizarla para cubrir sus necesidades analíticas sobre su sistema energética. Consideramos que el punto débil del sistema es el modelo de datos utilizado en los análisis; en el sistema construido hemos utilizado el modelo de datos utilizado por la Universidad Complutense de Madrid en la salida de datos de sus sensores energéticos, pero éste no tiene por qué corresponderse al modelo de otros sistemas. Por ello, el paso inmediatamente siguiente sería ajustar la flexibilidad del modelo de datos.

Por otra parte, el modelo de optimización podría ser mejorado principalmente poniéndolos a prueba en datos desconocidos; con el paso del tiempo podríamos coleccionar más datos a través de los sensores de los múltiples edificios del sistema energético de la Universidad Complutense de Madrid. De esta forma, podríamos conseguir que nuestros modelos fueran más robustos y así otorgarle de mayor precisión al sistema global. Además, podría ser beneficioso para el análisis aumentar la complejidad de la representación del coste. Nuestro modelo considera costes de mantenimiento estáticos. En un futuro también podría agregarle valor a la plataforma incluir esta funcionalidad a través de una interfaz más elaborada, algo que quedó fuera del proyecto por falta de tiempo. Por último, también resultaría en una mejora del análisis considerar la dinámica de las baterías y las fluctuaciones en los precios de compra/venta de energía a la red eléctrica.

Un aspecto común a todas las funcionalidades que ofrece la plataforma que consideramos que es de vital importancia a mejorar es la validación de uso de la misma. El desarrollo ha sido realizado considerando un entorno concreto que, como ocurría con el

modelo de datos, nos fuerza a necesitar testear las hipótesis sobre las que hemos basado el diseño de los requisitos del proyecto con más usuarios. Así, funcionalidades extra que no hemos considerado en este entorno serían tenidas en cuenta y agregadas al sistema si descubriéramos que son de importancia en otros entornos.

El proyecto se centra en ser considerado un EMS (Energy Management System), aunque completo estos sistemas componen muchas funcionalidades mas que podrían ser añadidas, el sistema cumple con las funciones de monitorización y medición, pero no la funcionalidad de controlar las cargas eléctricas de sus edificios. La gestión de un sistema energético no es solo realizar análisis, también permitir realizar acciones directas sobre el sistema energético desde el propio proyecto, controlando centralmente dispositivos como unidades de climatización y sistemas de iluminación en múltiples ubicaciones.

Un EMS suele tener una visión cercana al campo del IoT, con diferentes modelos de datos, dispositivos de extracción de esta información y arquitectura que la gestione, por tanto, es necesario tener un sistema previo que actúe y permita acceder a los datos desde el EMS de forma directa. Es interesante la construcción de esta capa o realizar la conexión directa del proyecto a la fuente de datos para que el modelo y informes se nutran directamente de los datos y no solo de históricos, además de obtener una monitorización en tiempo real, con ocupación y consumo que extraiga estos datos de los sensores y no con simulaciones específicas.



# Capítulo 8

## Conclusions and future work

### 8.1. Conclusions

Given the results obtained and the development of the project, we can determine several conclusions. The first one is the importance that an energy management system (EMS) has in any organization because of the knowledge that it is able to generate about how the energy is being used in its facilities, how it could save expenses and how it could contribute to help the planet by producing less  $CO_2$ . Nowadays, renewable energies are gaining a lot of importance, thus generating an added difficulty to the control and energy monitoring of a center. This makes the use of a system like this more essential.

The importance of creating free tools in the field of energy has also been noticed, since there are many options that make up most or all the features that an organization may need but are usually very expensive and licensed privately. It is important to have a platform that covers the needs of an organization and that allows the community to improve the system, something that has worked very well in other fields. The project's repository can be accessed by navigating to [www.github.com/albertopastormr/tfg](http://www.github.com/albertopastormr/tfg). The project has been published under a *GNU General Public License v3.0*.

The creation of this platform has to be done with a clear mentality from the user's point of view, since business projects tend to forget about checking the user experience and cause these industrial tools of this kind to be very difficult to use. The project shows how a system can be built that, showing a strong analytical component, is capable of giving the end user a simple, intuitive, useful and easy to use design and interface.

The platform built in the project demonstrates how the analytical needs of various profiles can be grouped and satisfied, having the functions that each one needs and using a security control so that each one can observe only the information for which they have access. This component, besides helping in aspects of data privacy, constitutes a profile within its interests, thus avoiding to saturate with information that the user does not need.

Knowing the need that each profile has, the project is able to compose historical analyses that, together with information on estimates of photovoltaic energy use, generates complete reports on critical aspects to study on the consumption and energy savings of

an organization. The system also allows to know which is the optimal configuration of a photovoltaic installation from the desired consumption and location, offering the best possible budget and combination of elements for the system. This factor is very useful for a technician because of the support offered in the decision making, without the disposition of the project, he should study all the combinations of the possible components that can be adjusted to the system. Finally, the platform allows monitoring a building, showing if the current energy status and consumption are within the parameters that have been stipulated for an energy installation. In addition, thanks to the union of the data of occupation of that installation, the platform provides an alarm system that allows to control the most critical or important metrics.

In conclusion, the project offers to the people in charge of the energy facilities of the Complutense University of Madrid a specific EMS with which to reduce the uncertainty in their decision making, monitoring the system, performing energy analysis of historical data and real time data. The simplicity of the platform, along with data management, demonstrate that, given the statement that proposed to solve the project, we have obtained an effective system.

## 8.2. Future work

The platform has been developed with flexibility and usability in mind so that, through its publication as free software, any institution, company or individual can use it to cover their analytical needs on their energy system. We consider that the weak point of the system is the data model used in the analyses; in the built system we have used the data model used by the Complutense University of Madrid in the data output of its energy sensors, but this does not have to correspond to the model of other systems. Therefore, the next step would be to adjust the flexibility of the data model.

On the other hand, predictive behavior models could be improved mainly by testing them on unknown data; over time we could collect more data through the sensors of the multiple buildings of the energy system of the Universidad Complutense de Madrid. In this way, we could make our models more robust and thus give more precision to the global system.

A common aspect to all the functionalities offered by the platform that we consider of vital importance to improve is the validation of its use. The development has been carried out considering a specific environment that, as it was the case with the data model, forces us to need to test the hypotheses on which we have based the design of the requirements of the project with more users. Thus, extra functionalities that we have not considered in this environment would be taken into account and added to the system if we discovered that they are of importance in other environments.

The project is focused on being considered an EMS (Energy Management System), although complete these systems are many more features that could be added, the system meets the functions of monitoring and measurement, but not the functionality of



controlling the electrical loads of their buildings. The management of an energy system is not only to perform analysis, but also to allow direct actions on the energy system from the project itself, centrally controlling devices such as air conditioning units and lighting systems in multiple locations.

An EMS usually has a vision close to the field of the IoT, with different data models, devices to extract this information and architecture to manage it, therefore, it is necessary to have a previous system that acts and allows access to data from the EMS directly. It is interesting the construction of this layer or make the direct connection of the project to the data source so that the model and reports are fed directly from the data and not only from historical data, besides obtaining a real-time monitoring, with occupation and consumption that extracts these data from the sensors and not with specific simulations.



# Contribuciones individuales

En esta sección analizaremos cuáles han sido las contribuciones de cada uno de los miembros del equipo al proyecto.

## Alberto Pastor Moreno

Dada la magnitud del proyecto, la colaboración ha sido muy relevante en todo momento del desarrollo del trabajo. Desde el inicio del proyecto en Septiembre de 2019 hemos mantenido rigurosamente reuniones semanales Iván y yo para así mantenernos informados en todo momento de qué hacía cada uno. Esto ha tenido un impacto muy positivo en el resultado, ya que múltiples veces hemos requerido de la ayuda del otro en nuestro trabajo. Además, todas las decisiones pertinentes al desarrollo han sido consensuadas entre los dos. De esta forma, hemos logrado que ambos estemos conformes con todos los aspectos del trabajo.

Hemos mantenido reuniones telemáticas asiduas con Jorge a lo largo de todo el proyecto. Ambos hemos asistido a estas reuniones aportando a las conversaciones por igual. Todas las comunicaciones por correo con Jorge las hemos redactado y revisado ambos.

El proyecto comenzó con el establecimiento de los objetivos del mismo, para lo cual contribuimos por igual. En este momento decidimos en conjunto que desarrollaríamos la plataforma web juntos, pero que la sección de monitorización en tiempo real la desarrollaría Iván y la sección de optimización multiobjetivo la desarrollaría yo.

Tras esto, comenzamos a documentarnos juntos sobre las posibilidades arquitectónicas para la plataforma web. Dado que *Python* era el lenguaje que mejor rendimiento aportaba en las funcionalidades que queríamos desarrollar, buscamos un *framework web* basado en este lenguaje, así como herramientas para la monitorización y optimización. *Django* fue la opción elegida por ambos como *framework web*. También decidimos en este proceso que usaríamos *Spark Structured Streaming* para la monitorización en tiempo real y *jMetalPy* para la optimización multiobjetivo. Configuramos juntos los parámetros de *Django* y desarrollamos las secciones auxiliares de la web, como son *help* y *login/logout* y el correspondiente sistema de gestión de usuarios y grupos.

Para facilitar el desarrollo y posterior despliegue, monté el sistema utilizando *Docker*. Para ello, desarrollé un fichero de configuración para una imagen de Django, otro para Selenium, otro para Spark y, finalmente, el volumen de datos compartido por todos los contenedores del sistema.

Una vez establecida la plataforma web, Iván se encargó de desarrollar la generación de informes mientras yo redactaba lo correspondiente a esta sección del proyecto. Durante el periodo de tiempo que comprendió este desarrollo, realizamos reuniones diarias para que así yo le pudiera ayudar con aquello que fuera necesario y él pudiera notificarme los avances que realizaba. Además, mientras redactaba la sección de generación de informes en la memoria, desarrollé la configuración *Docker* del proyecto. Cuando terminamos con estas tareas, yo comencé a documentarme en optimización multiobjetivo, el diseño de instalaciones fotovoltaicas y la herramienta *jMetalPy*. Mientras, Iván se documentó en monitorización en tiempo real, en gráficas dinámicas en tiempo real y en la herramienta *Spark Structured Streaming*.

Comencé a desarrollar esta sección del proyecto formulando el problema matemáticamente. Establecí los objetivos a optimizar, las variables de decisión, las variables auxiliares y las restricciones por desigualdad. Cambié el enfoque de la formulación dos veces tras contrastarlo con Jorge. En cada cambio simplemente cambié las ecuaciones que constituían el problema. Una vez la formulación estaba completa, comencé a desarrollar el modelo en *Python* con *jMetalPy*. Para contrastar el modelo desarrollado en código con las hipótesis planteadas en los requisitos y estado del arte, desarrollé varias gráficas que acompañarían al razonamiento del modelo en la memoria. Todo este desarrollo fue acompañado con la pertinente redacción de la memoria. El código del modelo y de las gráficas lo realicé en un *Jupyter notebook* para facilitar la iteración. Tras conseguir obtener un modelo que respondiera a los requisitos establecidos en la formulación, lo sometí a las pruebas expresadas en la sección de experimentación del modelo de optimización multiobjetivo. Documenté los resultados en la memoria. Por último, incluí el código del modelo en la plataforma web para que los usuarios pudieran acceder a esta funcionalidad a través de un formulario.

Al terminar el desarrollo del modelo, me dediqué a corregir todo lo documentado en la memoria y a agregar secciones nuevas como arquitectura y conclusiones. Iván y yo trabajamos juntos perfilando y revisando el documento para su entrega.

## Iván Fernández Mena

Al ser un trabajo realizado entre dos personas y conociendo la magnitud de los problemas identificados al principio, quedó claro que era necesaria una gran colaboración y comunicación entre ambos integrantes del equipo. Para llegar a los objetivos marcados era necesario un reparto de la tareas de forma equitativa, además de la revisión por parte del otro miembro para asegurar que las funcionalidades estuvieran completas en la entrega. Comenzamos con una fase de generación de contenido y análisis. En las reuniones mantenidas con el director del proyecto se establecieron posibles objetivos y se desarrolló una versión inicial de la memoria donde se expresaban cada uno de estos problemas a querer resolver, conociendo de forma detallada si tenían sentido para el proyecto a realizar.

Tras especificar las funciones, lo principal era construir una plataforma que fuera la base del *Energy Management System*. Dado que ambos conocíamos *Django*, nos propusimos trabajar juntos para realizar una plataforma sólida y estable, que se pudiera lanzar de forma cómoda con *Docker*. La estructura del proyecto con *docker-compose* causaba una metodología de trabajo que entendíamos ambos, ya que lo utilizamos a diario en nuestras

prácticas en empresa. Generamos una arquitectura básica usando el patrón modelo-vista-controlador, que fuera fácilmente ampliable con los módulos que deseásemos y usando el apoyado que nos ofrece la herramienta *Django* para facilitar mucho su implementación de nuevas funcionalidades.

Con la base de la arquitectura de la plataforma web ya formalizada teníamos que continuar generando contenido en la web para así añadir los roles y empezar con la funcionalidad de generación de informes de consumo con energía solar. Ya que nos era indiferente qué secciones desarrollaríamos cada uno, al final, nos decantamos porque yo me encargaría de la funcionalidad mientras Alberto introduciría contenido (texto e imágenes) e introducía el control de roles.

Antes de comenzar a desarrollar el generador de informes, comencé a investigar sobre librerías de visualización gráfica escritas en *Python* e investigar los datos de origen para establecer qué indicadores y qué gráficas se podían mostrar en los informes, siempre teniendo en cuenta los intereses de los diferentes perfiles energéticos. Una vez seleccionado *Plotly* como herramienta de visualización y con el conocimiento de los informes a mostrar, investigué cómo podríamos obtener los datos de energía solar de la plataforma europea *PVGIS*. Tomé la decisión de que finalmente realizar *web scraping* y extraer los datos necesarios directamente desde la web, decisión tomada por la ausencia de una API de acceso directo. La realización del *scraper* se hizo con la tecnología llamada *Selenium*, herramienta que no conocía y tuve que investigar a fondo. La extracción de datos fue compleja debido a la cantidad de detalles que había que tener en cuenta para poder acceder a cada una de las combinaciones de la web.

Tras generar un *scraper* que fuera capaz de extraer los datos que deseábamos, era necesario mantener alguna estructura de gestión de los datos. Para ello, usamos ficheros *YAML*. Estos archivos han sido usados para gestionar los centros de los que tenemos datos y de qué espacios temporal tenemos. De esta forma tenemos centralizada la información general de los datos disponibles. Debido a esta decisión, era necesario generar un controlador que gestionase todas estas peticiones. Con el *scraper* implementado y la estructura de gestión de información, usada posteriormente para el resto de funcionalidades, procedí a realizar cada una de las gráficas o indicadores. Busqué cada una de las gráficas que más valor analítico diese y utilicé tecnologías como *HTML*, *CSS*, *JavaScript* y *Beautiful Soup* para que la visualización fuera la mejor posible para el usuario final.

A continuación, tras varias valoraciones con el director, se realizaron varios cambios en los objetivos finales y se especificó el futuro del proyecto de forma más completa. Como consecuencia se estudió como se distribuiría el trabajo y qué tareas se podrían realizar en paralelo. Dada mi experiencia con las tecnologías de *Big Data* y la de Alberto de modelos de predicción, decidimos que yo me encargaría de la sección de datos en tiempo real usando *Spark Structure Streaming* y él de la creación del *modelo multiobjetivo*. Este detalle facilitó poder trabajar de manera paralela desarrollando más rápido las funciones, pero conllevó un esfuerzo extra en la comunicación para trabajar juntos apoyándonos en todo lo posible.

Como en la funcionalidad específica de monitorización es necesario generar datos en tiempo real simulados, completamente en el caso de ocupación y parcialmente en el caso

de los datos de consumo, fue necesario investigar qué se solía usar para realizar esta tarea. Aunque existan opciones muy elaboradas y completas para la simulación compleja, al no ser el foco del proyecto me centré en la generación con una distribución normal. De esta forma, obtuvimos datos de ocupación lo suficientemente variados y completos para poder probar el proyecto. Estos datos se generaron una única vez debido a que no iba a ser necesario realizarlo en ejecución. En el caso de los datos que son extraídos en tiempo real se debía simular la funcionalidad de un sensor. La mejor manera era realizar una serie de *scripts* de *Bash* y *python* que simulaban la extracción de cada una de las líneas del centro solicitado y enviaban por el *socket* dictado cada cierto tiempo. Esto fue llevado a cabo para ambos conjuntos de información, consumo y ocupación.

Con el servidor *socket* creado investigué la documentación oficial de *Spark Structure Streaming* y algún libro especializado en desarrollos en tiempo real. Desarrollé el extractor que actuaba de cliente del *socket* y el transformador de datos que extraía cada una de las líneas obtenidas por comunicación en el *socket*. Tras este paso, se transforman a un único *dataframe* que en última instancia leerían las gráficas de *Plotly* en el *dashboard*. Para poder generar estas aplicaciones de *Spark* era necesario ingresar en el *Docker* un *cluster* de *spark* y *yarn* con un nodo *master* y varios *workers*. La configuración de un *cluster* suele generar siempre complicaciones a la hora de especificar el número de *cores*, memoria y ejecutores, entre otros detalles, se tiene que tener en cuenta las aplicaciones que se van a ejecutar y la gestión de los recursos. Con una configuración de dos "*workers*" y modificando las características fui capaz de generar una arquitectura de datos lo suficientemente estable para el proyecto y los programas que era necesario ejecutar. Por último, con *Plotly* se construyeron las diferentes gráficas e indicadores. Aunque ningún elemento usado se parece a los usados en los informes, la experiencia del uso de esta librería ser rápido en el desarrollo y entendimiento de la documentación oficial de *Plotly*. Fueron necesarias varias inserciones de código escrito en lenguaje *Javascript* para controlar funcionalidades del tiempo real o de los avisos visuales en la plataforma web.

En el caso de la memoria se realizaron varias iteraciones según las indicaciones y correcciones de nuestro director. Este trabajo fue repartido de forma equitativa y intentando ser coherente con las funcionalidades que había desarrollado cada uno. Siempre que se finalizaba una sección, se marcaba en un tablón digital compartido. Tras esto, esta sección debía ser revisada por el otro miembro antes de ser enviada al director.

# Índice de figuras

3.1.	Gestión de datos tardíos en Spark Structured Streaming . . . . .	18
3.2.	Diagrama de flujo del algoritmo Evolución Diferencial . . . . .	22
4.1.	Ejemplo de gráfica con ventanas temporales interactivas . . . . .	28
4.2.	Diagrama de satisfacción de la demanda energética usando paneles solares, baterías y la red eléctrica. Este árbol de decisión se utiliza en cada instante temporal. . . . .	31
4.3.	Análisis del coste horario en función del número de paneles solares en el caso sin baterías. Los arcos con flecha indican la pendiente de la recta correspondiente. . . . .	33
4.4.	Coste diario en función del número de paneles. El coste está compuesto por hasta $N$ rectas y cambia de pendiente en cada número crítico de paneles. El arco con flecha indica la pendiente de la recta en un intervalo genérico $k$ . . . . .	35
4.5.	Series temporales para el ejemplo . . . . .	36
4.6.	Ejemplo de optimización del número de paneles y baterías. La figura central muestra un diagrama de contorno del coste para cada configuración de paneles y baterías. En cada esquina, se muestran configuraciones particulares con flechas apuntando a la configuración que representan: solo paneles solares, que vende toda la energía sobrante a la red; sin baterías ni paneles, que compra toda la energía; el caso aislado, que ni se compra ni se vende de la red; y el caso óptimo, que es una mezcla de los anteriores. . . . .	37
4.7.	Coste total para diferentes valores de $\xi_P = \frac{C_P}{V_P g_T C_e}$ . Todas las gráficas tienen $\xi_B = 0,5, \eta = 0,5$ . . . . .	37
4.8.	Coste total para diferentes valores de $\xi_B = \frac{C_B}{V_B C_e (1-\eta)}$ . Todas las gráficas tienen $\xi_P = 0,76, \eta = 0,5$ . . . . .	38
4.9.	Número de paneles solares y de baterías en función del coste diario normalizado . . . . .	38
4.10.	Funciones objetivos con el conjunto óptimo de Pareto. Se puede ver como conectan los valores óptimos considerando cada objetivo por separado. . . . .	39
5.1.	Componentes del sistema y sus interacciones tras una petición . . . . .	43
5.2.	Diagrama de secuencia de interacción entre componentes . . . . .	46
5.3.	Web scraping con Selenium sobre PVGIS para obtener datos históricos mensuales . . . . .	49
5.4.	Web scraping con Selenium sobre PVGIS para obtener datos históricos diarios . . . . .	49
5.5.	Web scraping con Selenium sobre PVGIS para obtener datos históricos horarios . . . . .	50
5.6.	Muestra de cantidad de cada número en los datos simulados de ocupación . . . . .	52

5.7. Diagrama de flujo de procesamiento de datos de monitorización en tiempo real	54
5.8. Diagrama de despliegue orquestado con Docker . . . . .	59
6.1. Captura de pantalla de la sección de inicio de sesión . . . . .	62
6.2. Formulario de generación de informes con diferentes parámetros seleccionables . . . . .	63
6.3. Formulario general que permite seleccionar el año, centro y localización a analizar . . . . .	64
6.4. Informe con tabla de datos y su correspondiente gráfica con ventanas temporales . . . . .	65
6.5. Muestra de cómo podemos estudiar datos al detalle y ayudar al usuario a sacar conclusiones . . . . .	66
6.6. Tabla con información sobre consumo fotovoltaico de un centro . . . . .	66
6.7. Gráfica que compara la energía activa, reactiva y máxima . . . . .	67
6.8. Gráficas que muestran el consumo en kWh y el coste en euros usando energía fotovoltaica a lo largo del tiempo . . . . .	68
6.9. Gráfico de barras temporales donde se visualiza el consumo en kWh y euros con y sin el uso de energía solar. . . . .	69
6.10. Gráfica que compara el consumo energético de un centro con el uso de baterías y energía solar. . . . .	69
6.11. Gráfica de consumo con energía solar y baterías en un espacio temporal pequeño. . . . .	70
6.12. Gráfica de consumo con energía solar y sin baterías en un espacio temporal pequeño. . . . .	70
6.13. Formulario que muestra todas las opciones del análisis de consumo, energía solar y baterías. . . . .	71
6.14. Series temporales utilizando los datos de ARQ. LOPEZ OTERO del 2019	71
6.15. Frente de Pareto con la solución óptima de Pareto comparando el coste diario de la instalación con la electricidad consumida de la red . . . . .	72
6.16. Funciones objetivos con el conjunto óptimo de Pareto para los datos históricos. El punto naranja es la solución de caso aislado. . . . .	72
6.17. Ejemplo de datos de origen con información de consumo energético . . .	74
6.18. Ejemplo de datos de origen con información de ocupación de un edificio .	74
6.19. Columnas necesarias y insertadas al dataframe base con las transformaciones hechas con Spark Structure Streaming . . . . .	75
6.20. Formulario de acceso al dashboard que continua la estética de la web . .	76
6.21. Vista general del dashboard para este caso de estudio . . . . .	76
6.22. Gráfica de líneas en tiempo real que muestra el consumo en kW . . . . .	77
6.23. Gráfica de barras en tiempo real que muestra la ocupación del centro . .	78
6.24. Los tres indicadores disponibles en el <i>dashboard</i> . . . . .	78
6.25. Indicador de control dentro de los parámetros normales establecidos por el técnico en el formulario . . . . .	79
6.26. Indicador que muestra cómo se ha superado el umbral por un consumo superior al esperado . . . . .	80
6.27. Ejemplo de alarma producida con el dato recibido a las 22:00PM . . . .	80



# Índice de cuadros

6.1. Valores de entrada usados en el modelo energético para problema de optimización. . . . .	73
6.2. Valores para diferentes soluciones en el caso de los datos de las instalaciones de ARQ. LOPEZ OTERO. . . . .	73



# Bibliografía

- [1] AENOR, “Certificación del sistema de gestión energética iso 50001.” <https://www.aenor.com/certificacion/eficiencia-energetica/eficiencia-energetica-50001>.
- [2] S. Fernando, J. Scott-Brown, O. Şerban, D. Birch, D. Akroyd, M. Molina-Solana, T. Heinis, and Y. Guo, “Open visualization environment (ove): A web framework for scalable rendering of data visualizations,” *Future Generation Computer Systems*, vol. 112, pp. 785 – 799, 2020.
- [3] S. P. Blog, “Top 5 features solar pv simulation software must have.” <https://solarpoweredblog.com/top-5-features-solar-pv-simulation-software-must-have/>.
- [4] A. Psaltis, *Streaming Data: Understanding the Real-Time Pipeline*. Manning, 2017.
- [5] S. Singh, M. Singh, and S. C. Kaushik, “A review on optimization techniques for sizing of solar-wind hybrid energy systems,” *International Journal of Green Energy*, vol. 13, no. 15, pp. 1564–1578, 2016.
- [6] M. Z. Bill Chambers, *Spark: The Definitive Guide: Big data processing made simple*. O’Reilly UK Ltd, 2018.
- [7] F. Apache, “Documentación oficial de spark structure streaming..” <https://spark.apache.org/docs/latest/Structureddd-streaming-programming-guide.html>.
- [8] L. R. Nair, S. D. Shetty, and S. D. Shetty, “Applying spark based machine learning model on streaming big data for health status prediction,” *Computers Electrical Engineering*, vol. 65, pp. 393 – 399, 2018.
- [9] M. J. Mayer and G. Gróf, “Techno-economic optimization of grid-connected, ground-mounted photovoltaic power plants by genetic algorithm based on a comprehensive mathematical model,” *Solar Energy*, vol. 202, pp. 210 – 226, 2020.
- [10] S. Hadji, J.-P. Gaubert, and F. Krim, “Theoretical and experimental analysis of genetic algorithms based mppt for pv systems,” *Energy Procedia*, vol. 74, pp. 772 – 787, 2015. The International Conference on Technologies and Materials for Renewable Energy, Environment and Sustainability –TMREES15.
- [11] A. Baruah, M. Basu, and D. Amuley, “Modeling of an autonomous hybrid renewable energy system for electrification of a township: A case study for sikkim, india,” *Renewable and Sustainable Energy Reviews*, vol. 135, p. 110158, 2021.
- [12] Y. Fernández, A. Tobar, D. Peluffo, T. Manosalvas, and R. Miranda, “Optimization-based algorithms applied in photovoltaic systems,” 11 2019.

- [13] D. Grygar and R. Fabricius, “An efficient adjustment of genetic algorithm for pareto front determination,” *Transportation Research Procedia*, vol. 40, pp. 1335 – 1342, 2019. TRANSCOM 2019 13th International Scientific Conference on Sustainable, Modern and Safe Transport.
- [14] D. Gómez-Lorente, I. Triguero, C. Gil, and A. Espín Estrella, “Evolutionary algorithms for the design of grid-connected pv-systems,” *Expert Systems with Applications*, vol. 39, no. 9, pp. 8086 – 8094, 2012.
- [15] S. Yuen, C. Chow, X. Zhang, and Y. Lou, “Which algorithm should i choose: An evolutionary algorithm portfolio approach,” *Applied Soft Computing*, vol. 40, 12 2015.
- [16] M. Lovay, G. Peretti, and E. Romero, “Aplicación del algoritmo evolución diferencial en un método de dimensionamiento para filtros bicuadráticos,” 2017.
- [17] S. Novus, “Eight pv design solutions - solar novus today.” [https://www.solarnovus.com/eight-pv-design-solutions\\_N10771.html](https://www.solarnovus.com/eight-pv-design-solutions_N10771.html).
- [18] A. I. Grimaldo and J. Novak, “User-centered visual analytics approach for interactive and explainable energy demand analysis in prosumer scenarios,” in *Computer Vision Systems* (D. Tzovaras, D. Giakoumis, M. Vincze, and A. Argyros, eds.), (Cham), pp. 700–710, Springer International Publishing, 2019.
- [19] A. Benitez-Hidalgo, A. J. Nebro, J. Garcia-Nieto, I. Oregi, and J. D. Ser, “jmetalpy: A python framework for multi-objective optimization with metaheuristics,” *Swarm and Evolutionary Computation*, vol. 51, p. 100598, 2019.
- [20] F. Al-Turjman, C. Altrjman, S. Din, and A. Paul, “Energy monitoring in iot-based ad hoc networks: An overview,” *Computers Electrical Engineering*, vol. 76, pp. 133 – 142, 2019.
- [21] “Precio de la tarifa de luz por horas en tiempo real, howpublished = <https://tarifaluzhora.es/>.”
- [22] “Precios actualizados de paneles y placas solares.” <https://kitdeenergiasolar.com/placas-solares/precios>.
- [23] Selectra, “Baterías solares: Dónde comprar, precios y tipos de baterías.” <https://selectra.es/autoconsumo/info/componentes/baterias-solares>.
- [24] R. O. Bawazir and N. S. Cetin, “Comprehensive overview of optimizing pv-dg allocation in power system and solar energy resource potential assessments,” *Energy Reports*, vol. 6, pp. 173 – 208, 2020.
- [25] A. Slowik and H. Kwasnicka, “Evolutionary algorithms and their applications to engineering problems,” *Neural Computing and Applications*, vol. 32, pp. 12363–12379, Aug 2020.
- [26] W. Inoubli, S. Aridhi, H. Mezni, M. Maddouri, and E. Mephu Nguifo, “A comparative study on streaming frameworks for big data,” 08 2018.

- [27] Z. S. Yaghoobi, M. Esmail Nazari, and F. Safdarian, “Comparison between the optimal sizing of a photovoltaic system in interconnected and isolated modes using gravitational search algorithm,” in *2018 1st International Conference on Advanced Research in Engineering Sciences (ARES)*, pp. 1–6, 2018.
- [28] D. Gómez-Lorente, I. Triguero, C. Gil, and O. Rabaza, “Multi-objective evolutionary algorithms for the design of grid-connected solar tracking systems,” *International Journal of Electrical Power Energy Systems*, vol. 61, pp. 371 – 379, 2014.
- [29] A. Fichera, R. Volpe, and E. Cutore, “Energy performance measurement, monitoring and control for buildings of public organizations: Standardized practises compliant with the iso 50001 and iso 50006,” *Developments in the Built Environment*, vol. 4, p. 100024, 2020.
- [30] J. Malinauskaite, H. Jouhara, B. Egilegor, F. Al-Mansour, L. Ahmad, and M. Pusnik, “Energy efficiency in the industrial sector in the eu, slovenia, and spain,” *Energy*, vol. 208, p. 118398, 2020.
- [31] M. Azaza, A. Eskilsson, and F. Wallin, “An open-source visualization platform for energy flows mapping and enhanced decision making,” *Energy Procedia*, vol. 158, pp. 3208 – 3214, 2019. Innovative Solutions for Energy Transitions.



# Apéndice A

## Código fuente

### A.1. Optimización multiobjetivo con JMetalPy

```
def hour_cost(q,gxp,d,Xb,Ce,Cv):
    if gxp > d: # is the generation higher than the demand?
        s = gxp - d
        if s > Xb-q: # is the overgeneration capable of filling the batteries
            qn = Xb
            dc = -Cv*(s-(Xb-q))
        else:
            qn = q + s # load the batteries
            dc = 0
    else:
        if d - gxp > q: # is the leftover demand higher than the load in the batteries?
            qn = 0
            dc = Ce*(d-gxp-q)
        else:
            qn = q - (d-gxp)
            dc = 0
    return dc, qn

def electricity_cost(Xp,Xb,gv, dv, N, Ce, Cv, qinit):
    c = 0
    q = qinit
    if q>Xb:
        q = Xb
    qv = []
    for i in range(N):
        [dc, q] = hour_cost(q, Xp*gv[i], dv[i], Xb, Ce, Cv)
        c += dc
        qv.append(q)
    return c, qv

def total_cost(Xp,Xb,irradiance, demand, N, Ce, Cv, qinit, Cb, Vb, Cp, Vp, Ninit = 2):
    ecost, battery_charge = electricity_cost(Xp,Xb,irradiance,
                                             demand, N, Ce, Cv, qinit)
```

```
for i in range(Ninit):
    ecost, battery_charge = electricity_cost(Xp,Xb,
                                             irradiance, demand,
                                             N, Ce, Cv, battery_charge[-1])

fixcost = Xb*Cb/Vb + Xp*Cp/Vp
npe_rv = np.asarray(e_rv)
rcost = npe_rv[npe_rv> 0].sum() # electricity from the network
total_cost = ecost + fixcost
return total_cost, rcost
```



Iván Fernández Mena y Alberto Pastor Moreno  
Septiembre 2020  
Ult. actualización 18 de septiembre de 2020  
L<sup>A</sup>T<sub>E</sub>X lic. LPPL & powered by **TEF<sub>L</sub>ON** CC-ZERO

Esta obra está bajo una licencia Creative Commons “CC0  
1.0 Universal”.

